

أوراكل



المركز الرئيسي: 11 شارع د /محمد رافت – محطة الرمل – الإسكندرية
تليفون وفاكس: 4838326 (03)(2+)
موبايل : 0101634294 (2+) - 0123357844 (2+)
Email: info@egyptbooks.net
URL: www.egyptbooks.net

المادة العلمية
أسامة محمد فتحى
صياغة لغة عربية
قسم البحوث والتطوير التكنولوجى بدار البراء

رقم الإيداع

2007 / 25138

إهداء

إلى كل القراء العرب



الفصل الأول

تصميم قواعد البيانات فى أوراكل

Oracle Design for Data

مفهوم قواعد البيانات

قواعد البيانات تعنى تجميع كمية كبيـــرة مـــن البيانـــات ، أو المعلومات، ومن ثم عرضها بطريقة محددة ومنظمـــة؛ لكـــى يمكـــن الإستفادة منها.

وهى تشمل ما يلى:

1. إضافة المعلومات – أو البيانات – إلى الملف
2. حذف البيانات السابقة.
3. تغيير البيانات الحالية.
4. ترتيب البيانات وتنظيمها داخل الملفات.
5. عرض البيانات على هيئة نماذج ، أو تقارير.

قاعدة بيانات أوراكل

لاتعد الاوراكل لغة برمجة مستقلة ، لكنها لغة تـــستخدم فـــى برمجة قواعد البيانات؛ لذا ينبغى معرفة الفرق بين لغة قواعد بيانـــات الاوراكل، وبين لغات البرمجة الأخرى.

فهذه اللغة تشبه إلى حد قريب (Microsoft Access) ، كما أنها قريبة من (Microsoft SQL Server) ، أى أنها قاعدة بيانات ، وليست لغة برمجة مستقلة بذاتها. فهى ليست مثل الفيجوال بيسك التى تعد إحدى لغات البرمجة ؛ لأنها تمكنك من عمل برامج ذوات أهداف

متعددة ، لاتقتصر على برامج قواعد البيانات ، إذ يمكنك من خلال الفيجوال بيسيك أن تقوم بعمل برنامج رسم ، كما هو الحال فى برنامج (الرسام) فى (ويندوز) ، كما يمكنك التحكم فيها بمرونة أكبر ، وتسمح لك إمكاناتها بالإرتباط بقواعد البيانات المختلفة.

إلا ان (أوراكل) تتميز عن غيرها بوجود ميزات عالية ، نذكر منها:

1. أنها تعتبر قواعد بيانات ضخمة ، مقارنةً بغيرها.
2. أنها تتمتع بأمان عال جدا ، وهذا سبب إنتشارها ، رغم تكلفتها العالية.
3. أن بها أدوات مساعدة للتعامل معها ، وإظهارها في أشكال عدة تعرف (تطبيقات أوراكل) ؛ أو الـــ (DEVELOPER) حيث يمكنك إدخال البيانات واستخراجها من خلال نماذج وتقارير ورسوم بيانية. غير أنها لا تتعامل مع قاعدة بيانات أخرى كما أنها لا تمكنك من عمل برامج مثل برنامج الرسام.

وهناك تخصصان للراغبين فى دراسة أوراكل :

الأول : إدارة قواعد بيانات أوراكل (Administration) .

والثانى: مطور قواعد بيانات أوراكل (Developer) .

وبما أننا نهدف فى هذا الكتاب إلى تعلم تطوير قواعد البيانات فلابد أن يكون الدارس متقنا لغة (SQL) و (PL/SQL) ، ثم نبدأ بعد ذلك فى معرفة الـ (Developer) ، من خلال الجزئية (Form) و (Report).

وسوف نبدأ الآن بالمبادىء الأساسية ، الخاصة بـــ (SQL) و (PL/SQL).

لغة الإستعلام الهيكلية Structured Query Language

هى عبارة عن مجموعة من الأوامر ، التى تحتاج البرامج اليها ، كما يحتاجها المستخدم للوصول إلى البيانات الموجودة داخل قاعدة بيانات أوراكل.

وفى منتصف السبعينيات كانت هذه اللغة تعرف بإسم (System R) ، ثم تم تطويرها بمعرفة شركة (IBM) آنذاك بعد أن كانت مجرد نموذج لنظام إدارة قواعد بيانات علائقية .

وفى عام 1976 تم توصيف لغة (SQL) فى مجلة (INM Journal of R&D) تحت اسم (SEQUEL2) ، ثم فى العام 1979 قامت شركة (Oracle) بطرح أول نسخة تجارية من لغة (SQL).

وهذه اللغة غير إجرائية (Non-Procedural Language) لأنها تتعامل مع مجموعة سجلات في وقت واحد، لا مع سجل وحيد ، كما أنها تساعد فى استكشاف البيانات بطريقة تلقائية.

وتحتوي لغة (SQL) على تعليمات تفيد المستخدم ، وتمكنه من إدارة النظام ، وعمل التطبيقات اللازمة ، كما تحتوي على مجموعة من الأوامر لإنجاز المهام المختلفة ، كالبحث عن البيانات ، والتعامل مع القواعد والسجلات ، فى تناسق وتكامل .

بيئة SQL*PLUS :

قبل درس (SQL) عادة بشرح أوامر لغة معالجــة البيانــات (Data (DML) Manipulation Language) ، خاصة الاستعلام عن البيانات و استعراضها.

لكننا فى – هذا الكتاب – سنحاول أن تكون خطواتنا حسب منهاج (أوراكل) ، فنبدأ بشرح قوائم اللغة ذتها ، وأوامر التحرير والملفات ، ليكون الأساس الذى سنبنى عليه متينا ، يتوافق مع منهجية أوراكل.

ولكى نبدأ فى دراسة هذا الأساس سنقسمه إلى درسين: نتحدث فى أولهماعن قوائم (SQL*PLUS) ، ومنح صلاحية الاتصال

بقاعدة البيانات ، وثانيهما نستعرض فيه أوامر القوائم التحرير والملفات.

فعند تشغيلك (SQL*PLUS) تظهر لك نافذة تسألك عن اسم المستخدم ، وكلمة المرور ، وتعريف اسم قاعدة البيانات ، وذلك في حالة ما إذا كنت تستخدم النسخة (Database) ، أما إذا لم يكن لديك أكثر من قاعدة بيانات فى الوضع الافتراضي عند التشغيل للمرة الأولى فتتم اسم المستخدم وكلمة المرور. مع ملاحظة أنه فى حالة (كمبيوتر الشبكة) يستحسن كتابة اسم قاعدة البيانات. ويمكنك استخدام اسم مثل (System) ، (Sys) للعمل بها ، حتى تقوم بإنشاء مستخدم خاص بك .

إنشاء مستخدم جديد :

لكى يمكنك إنشاء مستخدم جديد ، فإن ذلك يتم من خلال لغة (DCL) ، فهي المسئولة عن ذلك باستخدامها الأمرين: (revoke) و(grant . وسوف تحتاج إلى الأمر (Grant) فى حالتنا هذه . أما فى الدروس المتقدمة فسوف يتم استخدام الأمر (Revoke) ؛ لإعطائه صلاحيات العمل .

وفيما يلى خطوات إنشاء مستخدم جديد (Revoke) .

- أكتب أولا الكود التالى :

```
SQL> create user Nesma identified by osama ;
```

وسوف تظهر لك (رسالة خطأ) ، وذلك لعدم وجود صلاحيات تنفيـذ مثل هذا الأمر ، لذا ..

- عليك الدخول بالمستخدم (System) :

```
SQL> connect
Enter user-name: system
Enter password: *****
Connected.
SQL>
```

- الآن.. قم بتنفيذ أمر إنشاء المستخدم مرة أخرى . ثم أعطه صلاحية الإستخدام :

```
SQL> grant connect to Nesma;
```

وهكذا تكون أنشأت مستخدما بكلمة سر محددة ، ومنحته حق الإتصال بالنظام .

- إذن .. جرب هذه الصلاحية ، على النحو التالى :

```
SQL*Plus: Release 10.2.0.1.0 - Production الجمعة مارس 23
15:31:08 2007

Copyright (c) 1982, 2005, Oracle.  All rights reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 -
Production
With the Partitioning, OLAP and Data Mining options

SQL> connect
Enter user-name: system
Enter password: *****
Connected.
SQL> create user Nesma identified by osama ;

User created.

SQL> grant connect to Nesma;

Grant succeeded.

SQL> connect
Enter user-name: Nesma
Enter password: *****
Connected.
SQL> connect
Enter user-name: system
Enter password: *****
Connected.
```

أوامر التحرير في SQL*Plus :

أمر الإضافة (Append):

سنضيف (sql>A +).

فإذا كانت لدينا التعليمية الآتية – على سبيل المثال :

```
sql>select ename from
```

وأردنا إضافة الجدول (Emp) إلى الجملة السابقة ، بعد كلمة (Form) ، فإننا نضيف إليها ما يلى:

```
Sql>A emp ;
```

أمر التعديل (Change):

هذا الأمر يسمح بتغيير كلمةٍ تكون ضمن (التعليمة) بغيرها . وهو ما يفيدك عند كتابة اسم الجدول أو الحقل بشكل خاطيء ، لأنك في بيئة (SQL*PLUS) لا يمكنك الرجوع إلى السطر السابق في حالة كتابة التعليمات .

والشكل العام للتعليمة هو :

```
sql> C / old /new
```

وهذا هو الشكل القياسي لها ، حيثُ تعنى كلمة (Old) الكلمة المراد استبدال غيرها بها ، وتعنى كلمة (new) الكلمة الجديدة التى ستضاف.

فإذا كانت لدينا العبارة التالية:

```
sql>select enamee from emp dept
```

ثم أردنا تغيير كلمة (Enamee) إلي (Ename)
فالأمر يصبح هكذا :

```
sql> c / enamee / ename
```

وهناك صيغة أخرى يمكن استخدامها ، عندما نرغب فى حذف أحد النصوص ، وذلك من خلال كتابة الأمر التالى :

```
sql>C[HANGE] / text /
```

فلو كانت لدينا العبارة التالية – على سبيل المثال :

```
sql>select ename from emp dept ;
```

وأردنا حذف الكلمة (Dept):

فإننا نكتب :

```
sql> c / dept /
```

لتصبح العبارة بعد الحذف :

```
sql>select ename from emp ;
```

كيفية مسح مخزن الجمل :

ذكرنا في الدرس السابق أن أى (تعليمة) من تعليمات (SQL) يتم تخزينها في ذاكرة مؤقتة تسمي (SQL buffer) . وأنه بالإمكان مسح هذه الذاكرة بكتابة الأمر :

```
sql>clear buffer (cl buff);
```

ويمكن اختصار ذلك بكتابة ما بداخل القوس فحسب ، لتكون العبارة كالتالى :

```
sql>select * from tab ;
```

وهذا الأمر يقوم باستعراض أسماء جميع الجداول الموجودة في قاعدة البيانات ، فتظهر أسماؤها على الشاشة ، ويتك التخزين بشكل مؤقت فى (SQL buffer).

فإذا أردت مسح هذه الذاكرة أكتب الأمر التالي :

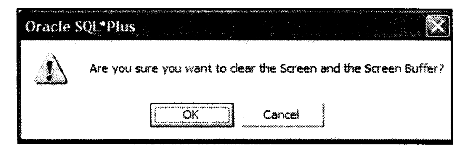
```
SQL> cl buff ;
```

عندئذ .. تظهر لك العبارة التالية ، التى تفيد بأنه تم مسح الذاكرة :

```
buffer cleared
```

هذا ..وهناك طريقة أخرى كالتالى :

- اضغط علي مفتاحي (shift +Delete) وستظهر لك الرسالة التالية :



- فاضغط علي (OK) ، لتتم عملية الحذف .

أوامر الملفات :

أمر حفظ الذاكرة فى الملف يكون على النحو التالى :

```
sql>sav file.sql
```

والمثال التالى يوضح ذلك أكثر :

```
SQL> connect
Enter user-name: scott
Enter password: *****
Connected.
SQL> select * from tab;

TNAME                          TABTYPE  CLUSTERID
------------------------------ ------- ----------
DEPT                           TABLE
EMP                            TABLE
BONUS                          TABLE
SALGRADE                       TABLE
AGC2006                        TABLE

SQL> select * from emp;

    EMPNO ENAME      JOB              MGR HIREDATE
SAL       COMM
---------- ---------- --------- ---------- -------- ----------
    DEPTNO
----------
```

```
  7369 SMITH     CLERK        7902 17/12/80      800

    20

  7499 ALLEN     SALESMAN      7698 20/02/81
1600      300
    30

  7521 WARD      SALESMAN      7698 22/02/81
1250      500
    30


  EMPNO ENAME     JOB          MGR HIREDATE
SAL      COMM
---------- ---------- --------- ---------- -------- ---------- ----------
  DEPTNO
----------
   7566 JONES     MANAGER       7839 02/04/81      2975
    20

   7654 MARTIN    SALESMAN      7698 28/09/81
1250      1400
    30

   7698 BLAKE     MANAGER       7839 01/05/81
2850
    30

EMPNO ENAME     JOB          MGR HIREDATE
SAL      COMM
---------- ---------- --------- ---------- -------- ---------- ----------
```

```
   DEPTNO
----------
    7782 CLARK      MANAGER       7839 09/06/81
2450
      10

    7788 SCOTT      ANALYST       7566 19/04/87       3000
     20

    7839 KING       PRESIDENT          17/11/81       5000
     10


   EMPNO ENAME      JOB           MGR HIREDATE
SAL      COMM
---------- ---------- --------- ---------- -------- ---------- ----------
   DEPTNO
----------
    7844 TURNER     SALESMAN      7698 08/09/81
1500         0
      30

    7876 ADAMS      CLERK         7788 23/05/87       1100

     20

    7900 JAMES      CLERK         7698 03/12/81        950

     30


   EMPNO ENAME      JOB           MGR HIREDATE
```

```
SAL      COMM
---------- ---------- --------- ---------- -------- ---------- ----------
   DEPTNO
----------
      7902 FORD      ANALYST        7566 03/12/81       3000
        20

      7934 MILLER    CLERK          7782 23/01/82       1300
        10


14 rows selected.

SQL>
```

ففى الأمر السابق قمنا بالدخول بإعتبارنا المستخدم (Scott) ، ثم عرضنا جداول قاعدة البيانات الموجودة ومن بعدها السجلات .

والآن يمكن حفظ هذه النتائج فى ملف نصى ، بكتابة الأمر التالى :

```
SQL> sav file.sql
Created file .sql
SQL>
```

أمر عرض محتويات ملف مسجل :

أمر عرض محتوى ملف مسجل يكون على النحو التالى :

```
sql>get file.sql
```

ولكن لاحظ أولا أننا سنقوم بكتابة (/) بعد المحث (SQL) لكى يتم تنفيذ الأمر المخزون فى الملف ، كما يلى :

```
SQL> get file.sql
  1* select * from emp
SQL> /
```

والمثال التالى يوضح الأمر :

```
EMPNO ENAME      JOB          MGR HIREDATE
SAL       COMM
---------- ---------- --------- ---------- -------- ---------- ----------
   DEPTNO
----------
      7369 SMITH      CLERK          7902 17/12/80        800
       20

      7499 ALLEN      SALESMAN       7698 20/02/81
1600        300
       30
```

```
  7521 WARD       SALESMAN        7698 22/02/81
1250       500
     30


  EMPNO ENAME      JOB            MGR HIREDATE
SAL      COMM
---------- ---------- --------- ---------- -------- ---------- ----------
  DEPTNO
----------

7566 JONES      MANAGER        7839 02/04/81      2975
     20

  7654 MARTIN     SALESMAN        7698 28/09/81
1250      1400
     30

  7698 BLAKE      MANAGER         7839 01/05/81
2850
     30


  EMPNO ENAME      JOB            MGR HIREDATE
SAL      COMM
---------- ---------- --------- ---------- -------- ---------- ----------
  DEPTNO
----------
  7782 CLARK      MANAGER         7839 09/06/81
2450
     10
```

```
    7788 SCOTT     ANALYST        7566 19/04/87      3000
     20

    7839 KING      PRESIDENT           17/11/81      5000
     10


   EMPNO ENAME      JOB              MGR HIREDATE
SAL      COMM
---------- ---------- --------- ---------- -------- ---------- ----------
   DEPTNO
----------
    7844 TURNER    SALESMAN       7698 08/09/81
1500        0
      30

    7876 ADAMS     CLERK          7788 23/05/87      1100
     20

    7900 JAMES     CLERK          7698 03/12/81       950
     30


   EMPNO ENAME      JOB          MGR HIREDATE
SAL      COMM
---------- ---------- --------- ---------- -------- ---------- ----------
   DEPTNO
----------
    7902 FORD      ANALYST        7566 03/12/81      3000
     20
```

```
   7934 MILLER     CLERK           7782 23/01/82      1300
     10


14 rows selected.

SQL>
```

فتنفيذ محتوي ملف مسجل يكون:

```
sql>star file.sql
```

أو :

```
sql> @ file.sql
```

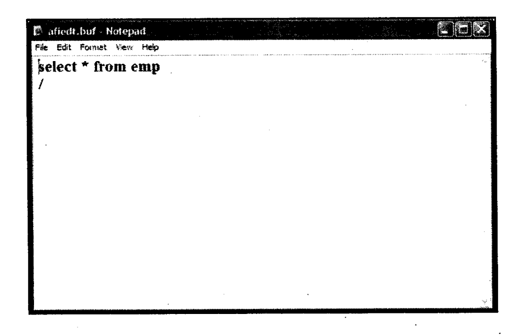
أما تشغيل برنامج التحرير فيكون :

```
SQL> ed
```

.. وهذا مثال يوضح ذلك :

```
SQL> ed
Wrote file afiedt.buf

  1* select * from emp
SQL>
```



ففى النافذة السابقة تظهر (التعليمية) التى سبق أن كتبناها فى الملف ، وبذلك يمكنك القيام بالتعديل ، أوالإضافة ، أوالحذف ، لأن مسح (تعليمة) تمت كتابتها في بيئة (SQL) لا يمكن القيام بها بعد تجاوزبسطر أو أكثر . لذلك يفضل البعض كتابة تعليمات (SQL) في المفكرة ، ثم نسخها بعد ذلك ولصقها في بيئة (SQL) . وهو أسلوب خاطئ يأتى نتيجة الكسل عن الكتابة فى الحال لكل تعليمية وتنفيذها في وقتها.

وهذا التفصيل خاطئ للأسباب التالية :

- أنه يجب عليك التعلم من أخطائك أثناء الكتابة.
- أنه يجب أن تعتاد على كتابة التعليمات.
- أنه ينبغي أن تتعرف على رسائل الخطأ ، لتتمكن من التعامل معها.

- ولكي تقوم بتحرير أحد الملفات ، وتحفظه تحت اسم معـــين ، عليك كتابة الأمر التالي :

```
SQL> ed ali.sql;
```

أمر إظهار بنية أحد الجداول :

هذا الأمر يهدف إلى إظهار صفات حقول الجدول ، من حيث النوع (رقم .. نص .. تاريخ) .. إلخ. كما يهدف إلى بيان حجمـــه ، وما إذا كان له قيد أم لا .

وصيغة هذا الأمر كالتالى :

```
SQL> desc tablename;
```

والمثال التالى يوضح ذلك :

```
SQL> desc emp;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------------------
----
 EMPNO                                     NOT NULL NUMBER(4)
 ENAME                                              VARCHAR2(10)
 JOB                                                VARCHAR2(9)
 MGR                                                NUMBER(4)
 HIREDATE                                           DATE
 SAL                                                NUMBER(7,2)
 COMM                                               NUMBER(7,2)
 DEPTNO                                             NUMBER(2)

SQL>
```

إنشاء الجداول في قاعدة البيانات :

عرفنا فيما سبق أن لغة (DDL) إحدى لغات (SQL) الفرعية ، وأنها تختص بتعريف البيانات ، كما أنها الأصل الذى تبنى عليه (SQL) .

فحين تقوم بتعريف هذه البيانات يتم إدخالها في قاموس البيانات الخاص بأوراكل.

و الأوامر الأساسية لهذه اللغة هي:

- CREATE TABLE: ويستخدم لإنشاء الجداول.
- ALTER TABLE : ويستخدم لتعديل جدول تم إنشاؤه سابقا .
- DROP TABLE : ويستخدم لحذف جدول غير مرغوب فيه.

أنواع أوامر إنشاء الجداول :

هناك نوعان من أوامر إنشاء الجداول :

أولا : إنشاء جدول جديد .

ثانيا : إنشاء جدول جديد ، ولكن من خلال خصائص جدول قديم .

وسوف نستعرض أولا إنشاء الجداول بنوعيها السابقين ، ثم نفصل الكلام عن كل نوع فيما بعد .

أما كود إنشاء الجداول فهو كالتالى :

```
CREATE TABLE اسم الجدول (حقول الجدول);
```

ولإسم الجدول شروط تنبغى مراعاتها ، هى :

1. ألا يتجاوز طول الاسم ثلاثين حرفا .
2. أن يبدأ بحرف لا برقم ، مع العلم بأن الاسم يمكن أن يكـــون خليطا من الأرقام والأحرف والرموز الخاصة .
3. ألا يكون اسم الجدول كلمة (محجوزة) فى اللغة .

كما أن هناك صفتان لحقول الجداول هما :ويجب أن نعلم أن لحقول الجدول صفتين هما:

طول الحقل : أي الحجم الذي سيخزنه في قاعدة البيانات .

القيود علي هذا الحقل : وهي تعني الشروط اللازمة لقيم هذه الحقول ، وسنتحدث عنها لاحقاً .

وأما البيانات فهى أنواع ، منها :

بيانات حرفية :

وهذا النوع يستخدم لتخزين عدد ثابت من الأحرف ، والحد الأقصى لعدد الأحرف لهذه الأحرف (2000 بايت) . ويكتب عدد الأحرف داخل الأمر :

```
CHAR (16)
```

بيانات حرفية كبيرة :

وهى تنقسم بدورها إلى نوعين :

الأول : (VarChar) ..

والثاني : (VarChar2) ..

فالأول يستخدم لتخزين بيانات حرفية متنوعة ، حدها الأقصى (4000 بايت) :

```
VarChar(50)
```

والثانى يستخدم لتخزين بيانات حرفية متنوعة ، حدها الأقصى (4000 بايت):

```
VarChar(50)
```

والفرق بين النوعين أن الأخير يدعى (المتغير المطاطى) ، لأننا إذا حجزنا عشر خانات ، فى حين كان الاسم يتكون من ست خانات ، فسوف يختصر – تلقائيا – إلى الخانات الست فى حين أن النوع الأول يقوم بحجز الخانات العشر جميعها ، حتى إذا لم تستخدم .

الحقل ذو القيمة الرقمية الصحيحة :

تتكون البيانات المدخلة في هذا الحقل من الأرقام ((0) ، (1) ، (2)) اختياري:

NUMBER (50)

الحقل ذو القيمة الرقمية الصحيحة ، مع الفاصلة العشرية :

وهذا الحقل تتكون الأرقام الداخلة فيه من ((1) ، (2)) :

NUMBER (I , j)

حيث يمثل (I) طول العدد العشري كاملاً، شاملاً العدد ، بينما (J) يمثل طول الأعداد العشرية، التى على يمين الفاصلة.

الحقل ذو القيمة الرقمية (Raw):

وهو يستخدم لتخزن البيانات الثنائية ، وأقصى طول له (2000 بايت).

الحقل ذو القيمة التأريخية (Date):

ويستخدم لتخزين البيانات من نوع التأريخ (يوم .. شهر .. سنة).

الحقل ذو البيانات الكبيرة :

ويستخدم هذا الحقل لخزن البيانات النصية ، التي يصل طولها إلى (2جيجا بايت) . وهناك أنواع أخرى منه ، ليست مدار اهتمامنا الآن، وهي :

- BLOB : كائن ثنائي كبير .
- CLOB : كائن كبير يعتمد على الأحرف .
- NCLOB : كائن كبير يعتمد على الأحرف وحيدة (البايت) أو متعددة البيانات.
- BFILE : ملف خارجي كبير.

والآن .. سنستعرض طريقة إنشاء جدول جديد ، وذلك بإمداد حقوله تعريفا من المتغيرات السابقة .

بداية العمل :

قبل البدء بإنشاء الجدول ادخل بالمستخدم الخاص بك ، ولا تخدل بالمستخدم (Scott) أو المستخدم (System) .

وقد سبق القول أن هناك طريقتين لإنشاء الجدول ، إحداهما الخاصة بإنشاء جدول جديد ، وثانيتهما إنشاء جدول جديد ، ولكن من خلال خصائص جدول قديم .

ولنبدأ بالطريقة الأولى ..

إنشاء جدول جديد :

سوف نقوم – الآن – بإنشاء جدول ، وليكن – على سبيل المثال – جدولا لدليل هواتف الشركات ، الذى سنضع له اسم (Phone Book) . وسوف يحتوى على الحقول الآتية .

- رقم السجل ، ونوعه(رقمي) ، بطول (10) مفتاح أساسي.
- اسم الشركة ، والهاتف ونوعه (حرفي) بطول (30) غير فارغ.
- رقم الهاتف ، ونوعه ، (رقمي) بطول (18) .
- تاريخ (تسجيل) السجل ، ونوعه (تاريخ) .

وذلك يستدعى الكود التالى :

```
CREATE TABLE PhoneBook
(
P_ID NUMBER (10) ,
P_Name VARCHAR2 (30) ,
P_Number NUMBER (18) ,
P_DATE DATE
) ;

CREATE TABLE PhoneBook(P_ID NUMBER (10) ,P_Name
VARCHAR2 (30) ,P_Number NUMBER (18) ,P_DATE
DATE) ;
```

وانظر المثال التالى :

```
Enter user-name: nesma
Enter password: *****
Connected.
SQL> CREATE TABLE PhoneBook
  2  (
  3  P_ID NUMBER (10) ,
  4  P_Name VARCHAR2 (30) ,
  5  P_Number NUMBER (18) ,
  6  P_DATE DATE
  7  ) ;

SQL> CREATE TABLE PhoneBook(P_ID NUMBER (10)
,P_Name VARCHAR2 (30) ,P_Number NUMBER (18)
```

```
,P_DATE DAT
E) ;
Table created.

SQL>
```

ويمكننا التأكد من خلال الأمر Desc :

```
SQL> desc phonebook;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------------------
----
 P_ID                                          NUMBER(10)
 P_NAME                                            VARCHAR2(30)
 P_NUMBER                                            NUMBER(18)
 P_DATE                                          DATE

SQL>
```

ثم لنأتى الآن إلى الطريقة الثانية ..

إنشاء جدول بخصائص آخر قديم :

سوف نقوم – الآن – بإنشاء جدول جديد، يحتوى على بعض الحقول ، دون أن نقوم بتعريف أطوالها ، لأننا سنستعين بصفات حقول أخرى نختارها من جدول آخر .

وللقيام بذلك استخدم الصيغة التالية :

```
SQL> CREATE TABLE FriendsPhoneBook
(F_ID,F_Name,F_Phone)
AS
SELECT P_ID, P_Name, P_Number
FROM phonebook ;
```

ولكنك ستجد أنك لم تتمكن من نشاء جدول ، والسبب فى ذلك عـــدم وجود صلاحيات لديك بذلك ..

وانظر المثال التالى :

```
SQL> connect
Enter user-name: nesma
Enter password: *****
Connected.
SQL> CREATE TABLE PhoneBook(P_ID NUMBER (10)
,P_Name VARCHAR2 (30) ,P_Number NUMBER (18)
,P_DATE DAT
E) ;
CREATE TABLE PhoneBook(P_ID NUMBER (10) ,P_Name
VARCHAR2 (30) ,P_Number NUMBER (18) ,P_DATE
DATE)
*
ERROR at line 1:
ORA-01031: insufficient privileges
SQL>
```

ولكى تحصل على تلك الصلاحية عليك تنفيذ الأمر التالى :

```
grant resource to Nesma ;

SQL> connect
Enter user-name: system
Enter password: *****
Connected.
SQL> grant resource to Nesma
  2  ;

Grant succeeded.

SQL>
```

القيود constraint

هناك طريقتان لإنشاء القيود . غير أن القيد (Default) لا يعمل على أى منهما . وهو ما سنشرحه بعد قليل .

وهاتان الطريقتان هما :

1. In line constraint :

وهو يعنى إنشاء قيود على مستوى تعريف الحقل ، أي أن القيد يكتب فى سطر تعريف الحقل نفسه ، أو فى السطر التالي مباشرة .

2. Out line constraint :

وهو يعني إنشاء قيود على مستوى تعريف الجدول ، أي إن القيود جميعها تكتب عقب الانتهاء من تعريف الحقول . وهذه الطريقة هي التى تعتمدها الشركة المنتجة ، كما أنها تعد الأفضل ، كما يرى الكثيرون .

القيد Not Null :

فيما يتعلق بالطريقة الأولى ، إليك هذا المثال :

```
SQL> create table customer (
2 cust_no number(4) not null,
3 cust_name varchar2(40)
not null,
4 cust_address varchar2 (30)
5);
```

وفى ذلك المثال السابق أنشأنا جدولا خاصا بالعملاء ، فالحقل الأول هو رقم العميل (cust_no) ، والحقل الثاني اسم العميل (cust_name) .

وكلاهما تم تعريفه بألا يكون ذا قيمة فارغة ، لأنه من البديهى أن تتم معرفة رقم العميل واسمه .

أما الحقل الثالث ، وهو عنوان العميل (cust_address) فلا يجب فيه معرفة رقم رقم العميل واسمه ، لذا ، فلابأس أن يترك فارغا عند إدخال البيانات .

ولاشك أنك لاحظت أننا ذكرنا اسم القيد (not null) عقب تعريف الحقل مباشرة ، سواء في السطر نفسه ، كما في حالة تعريف حقل رقم العميل ، أو في سطر آخر ، كما في حالة تعريف حقل اسم العميل .

وهناك مثال علي الطريقة الثانية :

```
SQL> create table customer (
2 cust_no number(4),
3 cust_name varchar2(40) ,
4 cust_address varchar2 (30),
5 constraint cust_no_nt
6 check (cust_no is not null),
7 constraint cust_name_nt
8 check (cust_name is not null)
9);
```

وهكذا ترى أننا عرفنا الحقول أولا ، ثم وضعنا القيود . وهذه الطريقة أفضل من الطريقة الأولى .

فعلى سبيل المثال ، حين عرفنا الحقل (رقم العميل) كتبنا الآتى :

```
constraint cust_no_nt
```

حيث أن كلمة (constraint) كلمة (محجوزة) ، معناها قيد ، ثم ذكرنا اسم القيد ، الذى يتكون من جزأين :

الأول .. هو اسم الحقل نفسه .

الثاني .. هو اختصار لاسم القيد ، حيث اختصرناه هنا بـــ (nt) ، وهكذا نكتب أسماء سائر القيود الأخرى .

أما check (cust_no is not null) فتعنى فحص الحقل (cust_no) للتأكد من أنه ليس فارغا ، حيث إننا نريده محتويا على قيمة .

القيد Unique :

هذا القيد يعنى ألا تتكرر قيمة الحقل الذى سوف نضع عليه ، فلو أننا وضعنا هذا القيد على حقل (رقم العميل) – على سبيل المثال – فمعنى ذلك أننا نريد ألا يتكرر رقم العميل ، وذلك على النحو التالى:

مثال ، وفقا للطريقة الأولى :

```
SQL> create table customer (
2 cust_no number(4) unique,
3 cust_name varchar2(40) not null,
4 cust_address varchar2(30)
5);
```

ومثال ، وفقا للطريقة الثانية :

```
SQL> create table customer (
2 cust_no number(4),
3 cust_name varchar2(40) not null,
4 cust_address varchar2(30),
5 constraint cust_no_uni
6 unique (cust_no)
7);
```

القيد Default :

هذا القيد يفيدنا في وضع قيمة افتراضية لحقل من الحقول ، فإذا كان الحقل يخص (عمر الطالب) ، وضعنا قيمة افتراضية لعمره، وذلك في حالة إذا لم يتم إدخال أي قيمة من قبل .

وهذا القيد هو الوحيد الذي يكتب بطريقة (In Line) .

وإليك المثال التالي :

```
create table student (
std_no number(7) not null,
std_name varchar2(40) not null,
std_age number(2)
default 20 ,
std_nation varchar2(20)
default 'Egyptian'
);
```

وكما تلاحظ ، فقد عرفنا حقل عمر الطالب (std_no) بإعتباره رقما ثم أعطيناه قيمةً افتراضية في حالة عدم إدخال أي قيمة ، وكانت القيمة 20 سنة .

كذلك عرفنا حقل جنسية الطالب (std_nation) على أنه نصي ، وأعطيناه قيمةً افتراضية في حالة عدم إدخال المستخدم أية قيمة ، وهي الجنسية (Egyptian).

القيد Check :

هذا القيد يفيدنا فى حالة الرغبة فى فحص قيمة مدخلة لحقل معين ، يقبل مجموعة قيم تم تحديدها من قبل ، حيث يقوم ذلك القيد بفحص القيمة المدخلة، من بين القيم الموجودة .

مثال : وفقا للطريقة الأولي :

```
SQL> create table student (
2 std_no number(7) not null,
3 std_name varchar2(40) not null,
4 std_sex varchar2(1)
5 check (std_sex in (' m ',' f ')),
6 std_case varchar2(1)
7 check (std_case in (' s ',' m',' w ',' d ')),
8 std_age number(2)
9 check (std_age between 19 and 30)
10);
```

هنا .. عرفنا حقل جنس الطالب (std_sex) على أنه (نصي) ، وأعطيناه قيد الفحص (Check) .

فحين يدخل المستخدم الحرف (M) فذلك يعنى أن (الطالب) ذكر ، أما إذا أدخل الحرف (F) فمعناه أن(الطالب) أنثى .

وعرفنا حقل الحالة الاجتماعية (std_case) على أنه (نصي) ، وأعطيناه قيد الفحص (Check) .ويتم الفحص بإدخال حرف لفحص قيمة الحقل:

- فإذا أدخل المستخدم الحرف (S) فهذا يعني أن الطالب متزوج.
- وإذا أدخل الحرف (W) فمعناه أن الطالب أرمل .
- وإذا أدخل الحرف (M) فمعناه أن الطالب مطلق .
- فإذا أدخل الحرف (D) فمعناه أن الطالب (أعزب) .

وعرفنا حقل عمر الطالب (std_ age) على أنه (نصي) وأعطيناه قيد الفحص (Check) .

وهنا يفحص قيمة الحقل القيمة العمرية من 19 إلي 30 .

مثال وفقا للطريقة الثانية :

```
SQL> create table student
2 (std_no number(7) not null,
3 std_name varchar2(40) not null,
4 std_sex varchar2(1),
5 std_case varchar2(1),
6 std_age number(2),
7 constraint std_sex_chk
8 check (std_sex in (' m ',' f ')),
9 constraint std_case_chk
10 check(std_case in (' s ',' m',' w ',' d ')),
11 constraint stdstd_age_chk
12 check (std_age between 19 and 30)
13);
```

القيد Primary Key:

هذا القيد وظيفتة إعطاء صفة (المفتاح) الرئيسى لحقل معين من عدة حقول في الجدول ما . وهذا المفتاح هو الذي يحدد بشكل وحيد ومتفرد، بحيث متميز عن غيره ، فلا تتكرر قيمته في أكثر من حقل واحد ، ولا يقبل قيمة (Null) ، أي إننا لا نستطيع ترك الحقل فارغاً ، دون قيمة .

وهذا مثال وفقا للطريقة الأولي :

```
SQL> create table student
2 (std_no number(10)
3 primary key,
4 std_name varchar2(50)
5);
```

ولاشك انك لاحظت أننا أنشأنا جدول الطلاب بمفتاح رئيسي هو حقل رقم الطالب (std_no) ، وعلى هذا الأساس نتعامل مع بيانات الطلاب ،من خلال هذا الحقل، الذي أصبح مفتاحا رئيسيا ، فيمكن من خلاله البحث عن طالب معين ، كما يمكن حذف طالب معين ، بمجرد إدخال رقمه .

وهذا المفتاح يكون رئيسياً في جدول الطلاب .. ويكون (أجنبيا) في الجداول الأخرى .

وهذا مثال ، وفقا للطريقة الثانية :

```
SQL> create table student
2 (std_no number(10) ,
3 std_name varchar2(50),
4 constraint pk _std_01
5 primary key (std_no)
6);
```

القيد Foreign Key :

هذا القيد يعطى حقلا معينا (من عدة حقول في أحد الجدوال) صفة المفتاح (الأجنبي) . وهو حقل (صفة) ، أو أكثر، حيث يستخدم للربط بين جدولين . وقد سمي (أجنبيا) ليس من الحقول الموجودة في الجدول ، أي أنه مجرد حقل أضيف إلى أحد الجدوال لربطه بجدول آخر .

و المفتاح (الأجنبى) يستخدم بمثابة مؤشر يقابل المفتاح الرئيسي. ويجب أن يكون المفتاح (الأجنبى) من نفس نوع بيانات (Data Type) المفتاح الرئيسي . فإذا كان المفتاح الرئيسي من النوع (رقم) – على سبيل المثال – فيجب أن يكون المفتاح الأجنبي من النوع (رقم) هو الآخر ، ولذلك يعد (الأجنبي) مؤشرا للرئيسي .

وهذا مثال وفقا للطريقة الأولي :

لكى يمكن إنشاء مفتاح أجنبى فى أحد الجداول ، يجب أن يكون هذا المفتاح رئيسياً في جدولٍ آخر سبق إنشاؤه .

وفي مثالنا هذا – سنقوم إن شاء الله – بإنشاء مفتاح رئيسي في جدول الأقسام ، وهو رقم القسم ، ثم نقوم بإنشاء حقل أجنبي في جدول الطلاب، يعود الى الرئيسي. وسوف يتضح ذلك من خلال عرض المثال .

1. إنشاء جدول الأقسام ، وهو الجدول الرئيسي (Master) :

```
SQL> create table section
2 (sec_no number(2)
3 primary key ,
4 sec_name varchar2(20) not null
5);
```

لاحظ أننا أنشأنا جدول الأقسام (وفيه رقم القسم (sec_no) بإعتباره مفتاحا رئيسيا ، واسم القسم ((sec_name) بقيد ليس فارغاً .

2. إنشاء جدول الطلاب ، وهو جدول (التفصيل) (Detail) :

```
SQL> create table student (
2 std_no number(7)
3 primary key,
4 std_name varchar2 (30) not null,
5 sec_no number(2) references
6 section (sec_no)
7);
```

لاحظ أننا أنشأنا جدول الطلاب ، وفيه رقم الطالب ، بإعتباره مفتاحا رئيسيا ، واسم الطالب بقيد لا يكون فارغاً .

كما أننا عرفنا الحقل ، واسمه رقم القسم (sec_no الأولي) ، وهذا أمر ضروري ، إذ يجب تعريف (الأجنبي) في الجدول بالتفصيل أولا، ثم يتم بيان أنه يشير إلى مفتاح رئيسي في جدول آخر، حجمه هو حجم رقم القسم في جدول الأقسام . لأن المفتاح (الأجنبي) يجب أن يكون تعريفه ، وحجمه من نفس نوع بيانات المفتاح الرئيسي وحجمه، وإن لم يكن من الضرورى أن يكون بالاسم نفسه .

وقد كتبنا كلمة (references) ، التى تعني أنه يشير إلى المفتاح الرئيسي في جدول الأقسام (Section) وهو المفتاح (sec_no الثانية).

ويجب الإنتباه إلى أن ما سبق كله ، كان بطريقة In line .

وفيما يلى نستعرض المثال السابق نفسه ، ولكن بطريقة Out Line:

سنقوم – الآن – بإنشاء الجدول الثاني (التفصيل) ، وفقا لهذه الطريقة. وقد اخترنا الجدول الثانى لأن الأول لا يعنينا فى شىء ، إذ لا يمكننا حذف جدول الأقسام ، إلا بعد حذف جدول الطلاب ، وذلك لوجود علاقة بينهما.

فلنبدأ إذن ..

- أكتب مايلى :

```
SQL> drop table section;
```

وستظهر لك (رسالة خطأ) تفيد بعدم إمكان حذف الجدول ؛ لارتباطه بحقل رئيسي فى جدول الطلاب .

- إذن .. قم بحذف جدول الطلاب – بهدف إنشائه من جديد بطريقة (Out Line) :

```
SQL> drop table student;
Table dropped.
```

- وأما جدول الأقسام فلن نحذفه ، بل سندعه كما هو . ولكى نعرف الجداول التى لدينا الآن . اكتب :

```
SQL> select *from tab;
```

والآن .. سنقوم بإنشاء جدول الطلاب ، كماسبق أن أنشأناه بطريقة (In Line) ، أما المفتاح (الأجنبي) ، فسوف نقوم بإنشائه – إن شاء الله – وفقا لطريقة (Out Line) .

```
SQL> create table student (
2 std_no number(7)primary key,
3 std_name varchar2 (30) not null,
4 sec_no number(2),
5 std_age number(2),
6 constraint fk_std_01
7 foreign key (sec_no)
8 references section (sec_no)
9);
```

لاحظ أن هذا الجدول يحتوى على مفتاح ثانوى ، وهو رقم القسم (std_no) – إضافة إلى وجود تعريف للمفتاح الرئيسى فى هذا الجدول . وذلك المفتاح الثانوي يسمح بتحديد الحقل (المرجع) في الجدول المرتبط به ، والذي يمكن أن يأخذ اسماً آخر .

ولكي يكون تعريف المفتاح الثانوي صحيحا ، يجب أن يكون الحقلان من نفس نوع البيانات ، ونفس الحجم(كما سبق أن ذكرنا) .

وعند تعريف المفتاح الثانوي نخبر (أوراكل) بأننا نريد إنشاء تكامل مرجعي بين الحقل رقم القسم (sec_no) في الجدول student والجدول Section . وهذا الأمر يمنع الحقل في جدول الطلاب من احتواء قيمة غير موجودة في العمود المرتبط به في جدول الأقسام .

فإذا كان لدينا جدولان : أحدهما رئيسي (Master) ، تعتمد عليه الجداول الأخرى في البيانات ، والثاني تفصيلي (Detail) ،

وبينهما علاقة بواسطة مفتاح (أجنبي) . ففى هذه الحالة لايمكننا حذف البيانات الموجودة في الجدول الأب (Master) .

ولحل هذه المشكلة توجد طريقتان :

الأولى : هى حذف البيانات فى الجدول الابن (Detail) ، ثم حذف البيانات في الجدول الأب (Master) . وتلك الطريقة هى الأسهل لكنها الأضعف .

```
SQL> delete from student ;
SQL> delete from section ;
```

والثانية : هى الأقوى والأمثل . فعند تصميم الجدول الابن (Detail) نضيف العبارة التالية ، بعد تعريف المفتاح الأجنبي :

```
On delete cascade
```

وعلى هذا فإننا نكتب الجدول الابن (Detail) وهو- هنا - جدول الطلاب على النحو التالى :

```
SQL> create table student (
2 std_no number(7)primary key,
3 std_name varchar2 (30) not null,
```

```
4 sec_no number(2),
5 std_age number(2),
6 constraint fk_std_01
7 foreign key (sec_no)
8 references section (sec_no)
on delete cascade
9);
```

وعند كتابة هذه العبارة عقب تعريف الحقل الثانوي ، لن يسمح أوراكل بحذف السجل المشار إليه من الجدول الأب ، وهو الأقسام (section) فحسب ، بل سيتوالى الحذف فى الجدول الابن أيضاً، وهو– هنا – جدول الطلاب (student).

إضافة الحقول وتعديلها في الجداول

بعد أن عرفنا كيفية إنشاء الجداول ، بحقولها المختلفة وقيودها، وعرفنا كيفية حذفها يتبقى لنا أن نعرف كيفية الإضافة إلى هذه الجداول وكيفية تعديلها .

والتعديل يكون بنية الجدول ذاتها ، لا في بياناته ؛ لأن التعديل في البيانات يتم من خلال اللغة الفرعية الثانية من (SQL) ، وهى (DML) .

وفيما يلى تفصيل ما سبق :

أمر تعديل الجدول :

هذا الأمر يمكننا من خلاله فعل ما يلى :

- إضافة حقل جديد إلى جدول سبق إنشاؤه .
- التعديل في صفة أحد الحقول .
- الإضافة – أو التعديل – في قيود تم إنشاؤها على عدة حقول من قبل .
- تأهيل القيود ، أو إعادة تأهيلها من جديد .
- حذف القيود من الجداول .

إضافة حقل جديد إلى جدول سابق الانشاء :

عندما نقوم بإنشاء الجداول ،وندخل اليها البيانات قد نحتاج – أحيانا – لإضافة حقل ، أو أكثر لهذه الجداول .

وهذا يتم باستخدام الصيغة التالية :

```
ALTER TABLE اسم الجدول ADD
فارغ أو غير فارغ أو قيود (طولها) نوع البيانات اسم الحقل
);
```

والصيغة السابقة تستخدم بناءً على الكلمة (ADD) . ولكى نفهم هذه الصيغة جيداً ، سوف نقوم بإضافة حقل (جنس الطلاب S_SEX (ذكر أو أنثى) .

وسنقوم الآن باستعراض الجدول ، وذلك من خلال الأمر التالى :

```
SQL> desc school;
```

فإذا لم يكن هذا الجدول موجوداً ، سوف تظهر لك العبارة التالية :

```
Object does not exist.
```

- إذن .. أكتب الأوامر التالية لإنشاء الجدول ، كما سبق أن تعلمت :

```
CREATE TABLE SCHOOL
(
S_NO NUMBER (10) ,
S_NAME VARCHAR2 (30) ,
S_TYPE CHAR (15) ,
S_LOC VARCHAR2 (30) ,
S_DATE DATE
) ;
```

• ثم استعرض خصائص الجدول :

```
SQL> DESC SCHOOL;
```

وسوف تظهر تلك الخصائص على النحو التالى :

```
Name Null? Type
------------------------------- -------- ----
S_NO NUMBER(10)
S_NAME CHAR2(30)
S_TYPE CHAR(15)
S_LOC CHAR2(30)
S_DATE DATE
```

• الآن.. أضف حقل (جنس الطلاب) – ذكر أو أنثى – وذلك بكتابة الأمر التالى :

```
alter table school add
(s_sex varchar2(10)
);
```

وسوف تظهر لك العبارة التالية ، التى تدل على أن الحقل الجديد تمت إضافته إلى الجدول :

```
Table altered.
```

• ولكى تتأكد من صواب هذه الخطوة استعرض خصائص الجدول كما تعلمت سابقا :

```
SQL> DESC SCHOOL;
```

وسوف تظهر لك موصفات الجدول كالتالى :

```
Name Null? Type
------------------------------- -------- ----
S_NO NUMBER(10)
S_NAME VARCHAR2(30)
S_TYPE CHAR(15)
S_LOC VARCHAR2(30)
S_DATE DATE
S_SEX VARCHAR2(10)
```

وهكذا .. ستجد أن الحقل الجديد S_SEX قد تمت إضافته إلى الجدول .

التعديل فى صفة أحد الحقول :

لاشك أنك تحتاج أحياناً الىلتعديل مواصفات أحد الحقول ، أو تغيير صفته . وهذا أمر ممكن ، ولكن من خلال شروط يمكن إدراكها، من خلال القاعدة العامة ، حيث إن بعض عمليات النقص

تؤدى إلى ايجاد المشاكل ، بينما عمليات الزيادة نادرا ما تسبب مشاكل .

وصيغة تعديل مواصفات حقلٍ ما ، في أحد الجداول كالتالي :

```
ALTER TABLE اسم الجدول Modify
الخاصية أو الخواص + اسم الحقل
) ;
```

فإذا أردنا – على سبيل المثال – تعديل خاصية الحقل الجديد S_SEX ، الذي أنشأناه منذ قليل مؤخراً من النوع VarChar2 إلى النوع CHAR ، فعليك كتابة الأمر التالي :

```
Alter table school modify
(s_sex char(10)
);
```

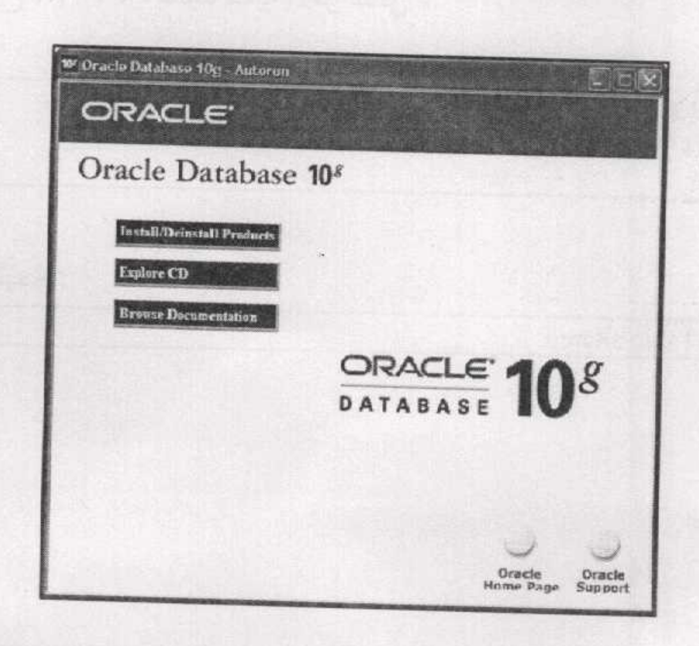
والنتيجة :

```
Table altered.
```

تركيب Oracle 10G :

يمكنك تحميل أخر (إصدارة) من أوراكل ، من خلال موقعها على الشبكة الدولية (الانترنت) ، وذلك بالبحث من خلال (Google). والأفضل أن تقوم بتحميل (الإصدارة) Oracle 10 Express Edition .

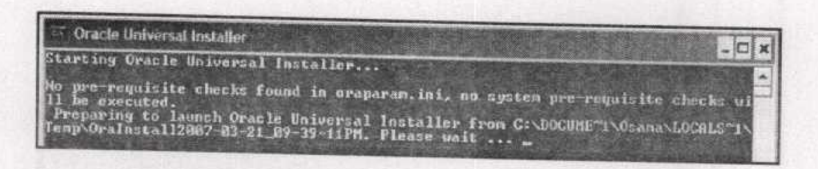
وعلى أية حال ، فسوف نقوم بشرح كيفية تركيب الإصدارة .. بعد قيامك بتحميل ملف (ISO) للأوراكل ، قم بحرقه باستخدام برنامج (NERO) ، علي (CD) ، أو استخدام Nero Image Drive لقراءة ملف الـــ (ISO) :



وعند تشغيل أسطوانة التركيب ستظهر لك النافذة السابقة .

- إذن .. اضغط علي Install/Deinstall Products :

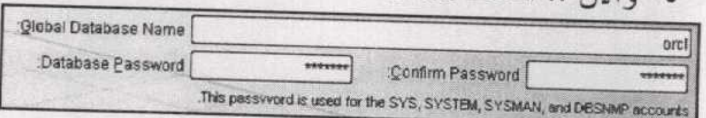
وسوف تظهر لك النافذة التالية لتستمر بضع ثوان :



ثم يظهر لك برنامج التركيب الرئيسي :

فاكتب اسم قاعدة البيانات فى الخانة Global Database Name والاسم الافتراضى لها (orcl) ، ولكن يمكنك كتابة اسم آخر ، كما يمكنك ترك الاسم الافتراضى كما هو . حسبما تريد .

- والآن .. اضغط Next :



وسوف سقوم البرنامج بالتأكد من أن النظام الذي تعمل وفقا له يقوم بتأمين كافة المتطلبات الأساسية اللازمة ، بحيث يعمل أوراكل عليه بشكل جيد . كما هو واضح فى الشكل التالى :

Oracle Universal Installer: Product-Specific Prerequisite Checks

Product-Specific Prerequisite Checks

The installer verifies that your environment meets all of the minimum requirements for installing and configuring the products that you have chosen to install. You must manually verify and confirm the items that are flagged with warnings and items that require manual checks. For details about performing these checks, click the item and review the details in the box at the bottom of the window.

Check	Type	Status
Checking Network Configuration requirements...	Automatic	Not executed
Validating ORACLE_BASE location (if set) ...	Automatic	Succeeded

Retry Stop

1 requirements to be verified.

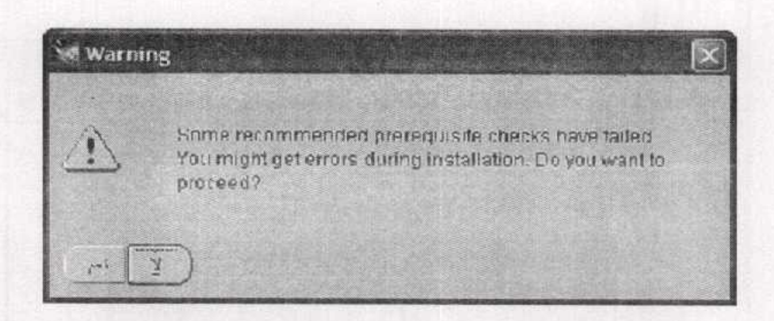
Check complete. The overall result of this check is: Passed

Checking Network Configuration requirements ...
Check complete. The overall result of this check is: Not executed <<<<

Help Installed Products... Back Next Install Cancel

ORACLE

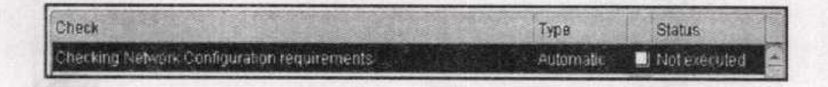
- إذن .. اضغط (Next) وسوف تظهر لك الرسالة التالية



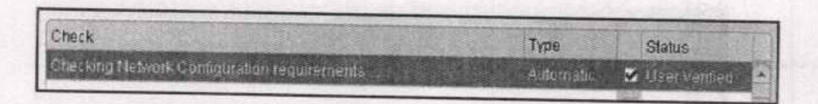
وتخبرك تلك الرسالة بعدم اكتمال مجموعة من التأكيدات (مثل اتصال الشبكة) فإذا كنت مهتما اضغط على (نعم) .. وإن لم تكن مهتما اضغط على (لا) .

ملاحظة :

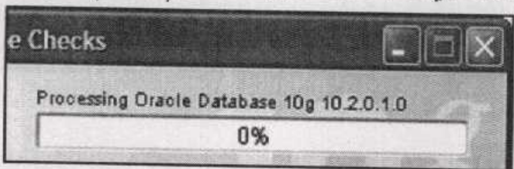
إذا اردت تأكيد هذا الاتصال اضغط علي الخيار التالى (الذى تراه فى النافذة السابقة على هذا الشكل) :



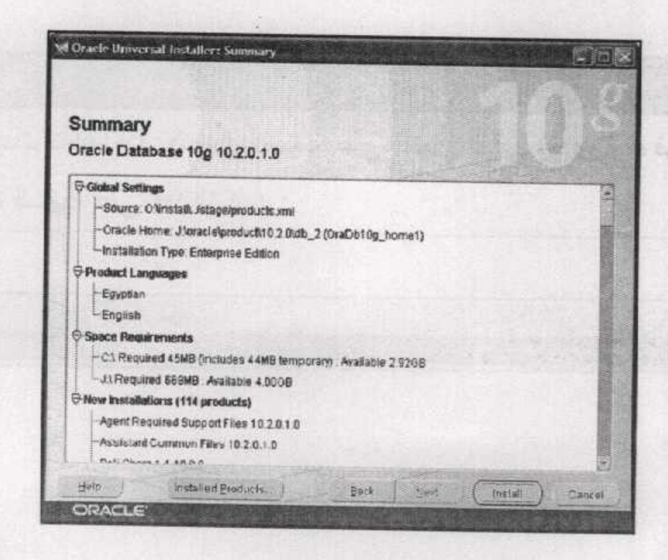
وسوف يتأكد برنامج التركيب من عمل الخاصية :



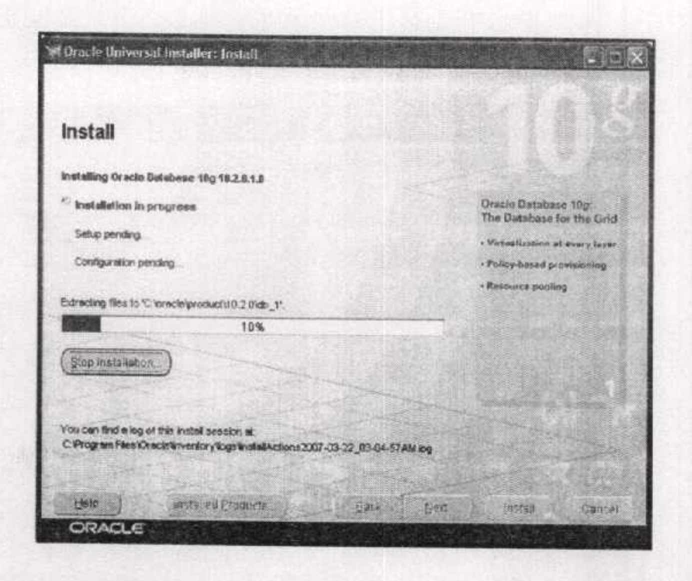
- ولبدء عمليات الضبط الآن اضغط (Next) : لبدء



- أما لبدء عمليات التركيب فاضغط علي (Install) ، كما هو واضح من الشكل التالى :



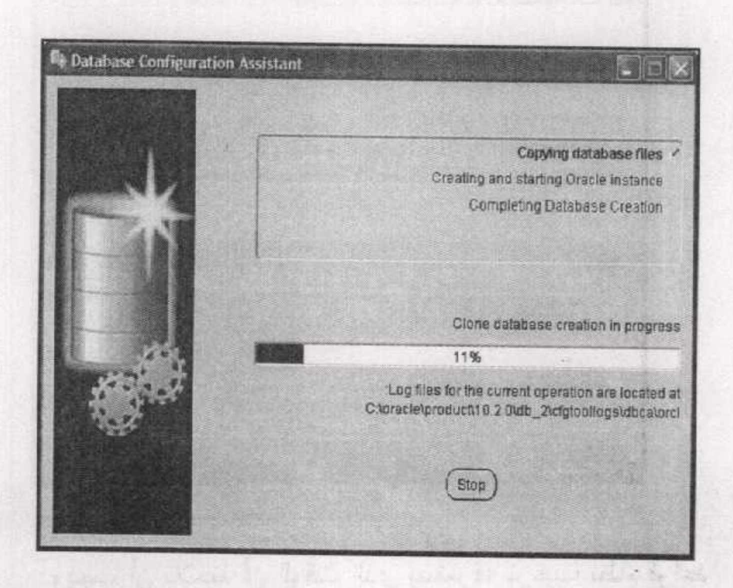
وسوف يبدأ البرنامج في تركيب الأوراكل علي النظام :



ويجب أن تلاحظ أن الوقت الذى يستغرقه تركيب نظام قواعد بيانات اوراكل يتوقف على مدى سرعة جهازك ، وكذلك النظام الذي يعمل عليه ذلك الجهاز ، فقد يستغرق ذلك وقتا يتراوح بين الثلاثين إلى ستين دقيقة .

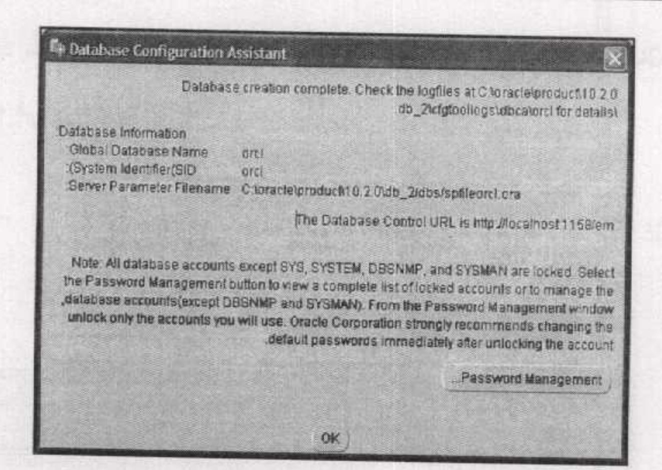
كما تجب ملاحظة أنه من الصعب تغيير بعض الإعدادات الافتراضية ؛ لذلك فمن الأفضل الحصول على نسخة لنظام (Image) من برنامج مثل (ghost).

الآن.. سيقوم (برنامج التركيب) ببناء قاعدة بيانات يمكنك أن تتدرب من خلالها :

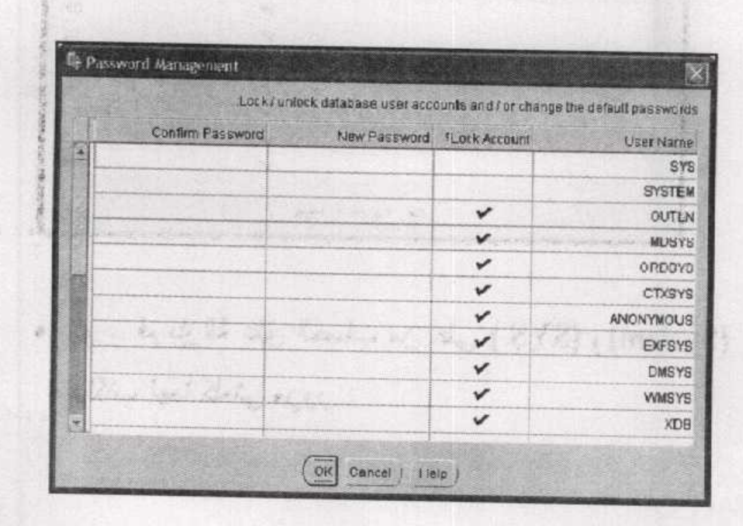


بعد الانتهاء من التركيب سيظهر لك البرنامج رسالة تفيد نجاح العملية ، ويطلب منك كتابة كلمة المرور .

- إذن .. اضغط علي الزر (Password Management) :

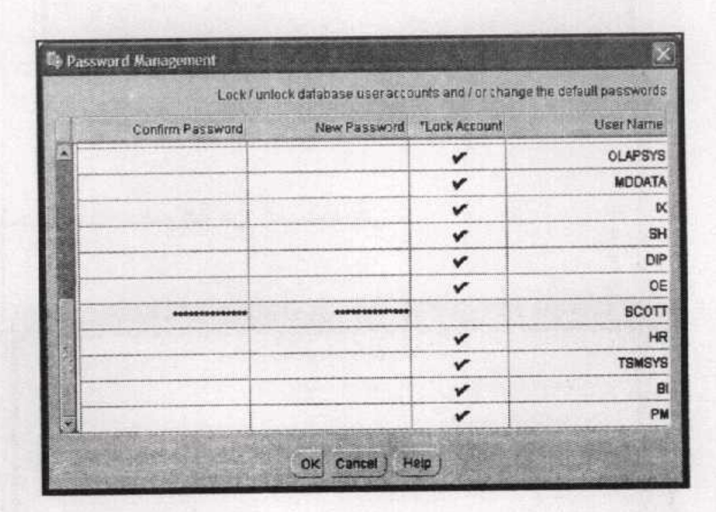


وستجد أمامك مجموعة من المستخدمين :

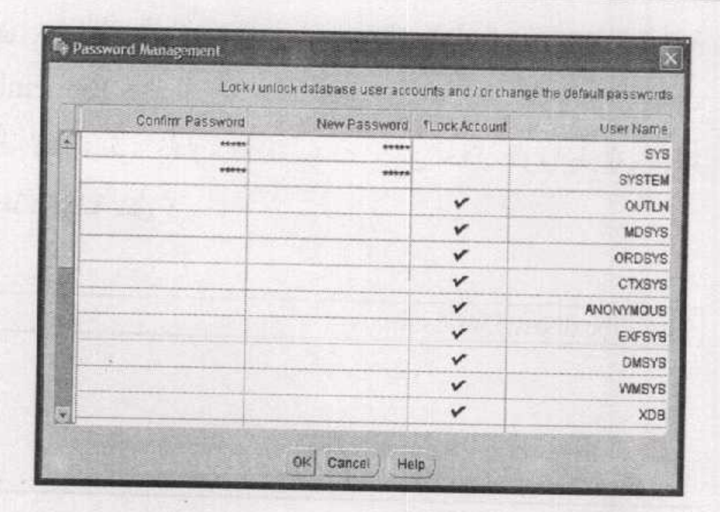


وستجد معظم هذه الحسابات مغلقة (Lock) ، لأنها مشار إليها بعلامة (√) ، في خانة (Lock Account) .

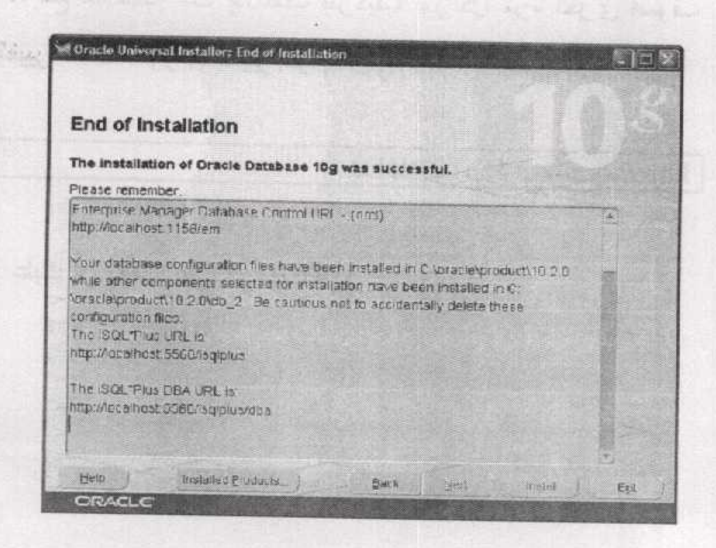
- إذن .. توجه إلي المستخدم (Scoot) ، وأزل علامة غلق الحساب ، ثم اكتب كلمة المرور :



- ثم .. قم بإزالة غلق الحساب من علي (SYS) و(System) واكتب لهما كلمتى مرور :



وهكذا تكون قد انتهيت من عملية التركيب .



ولكن .. عليك القيام بحفظ معلومات التركيب الموجودة في الخانة Please Remember ، وذلك قبل الضغط علي الزر (Exit) ، وذلك لأننا سنقوم بإدارة قاعدة بينات ، من خلال الروابط الموجودة فى هذه الخانة مثل :

```
http://localhost:5560/isqlplus
```

أو :

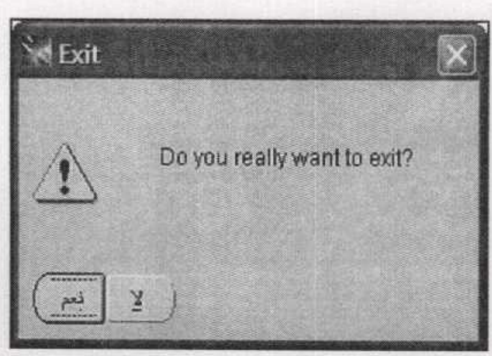
```
http://localhost:1158/em
```

هذا .. مع ملاحظة أنك إذا قمت بتركيب أوراكل مرة أخرى فسوف يتم تغيير المنفذ . وربما تغيير الرابط إلي :

```
http://localhost:5561/isqlplus
```

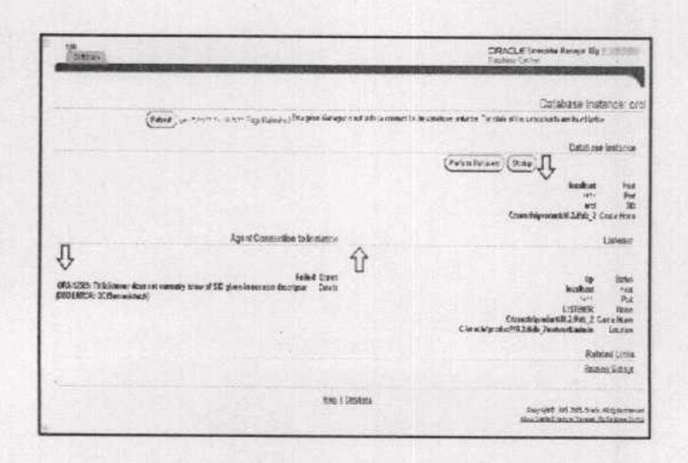
لذلك عليك أن تعمل على الرابط القديم

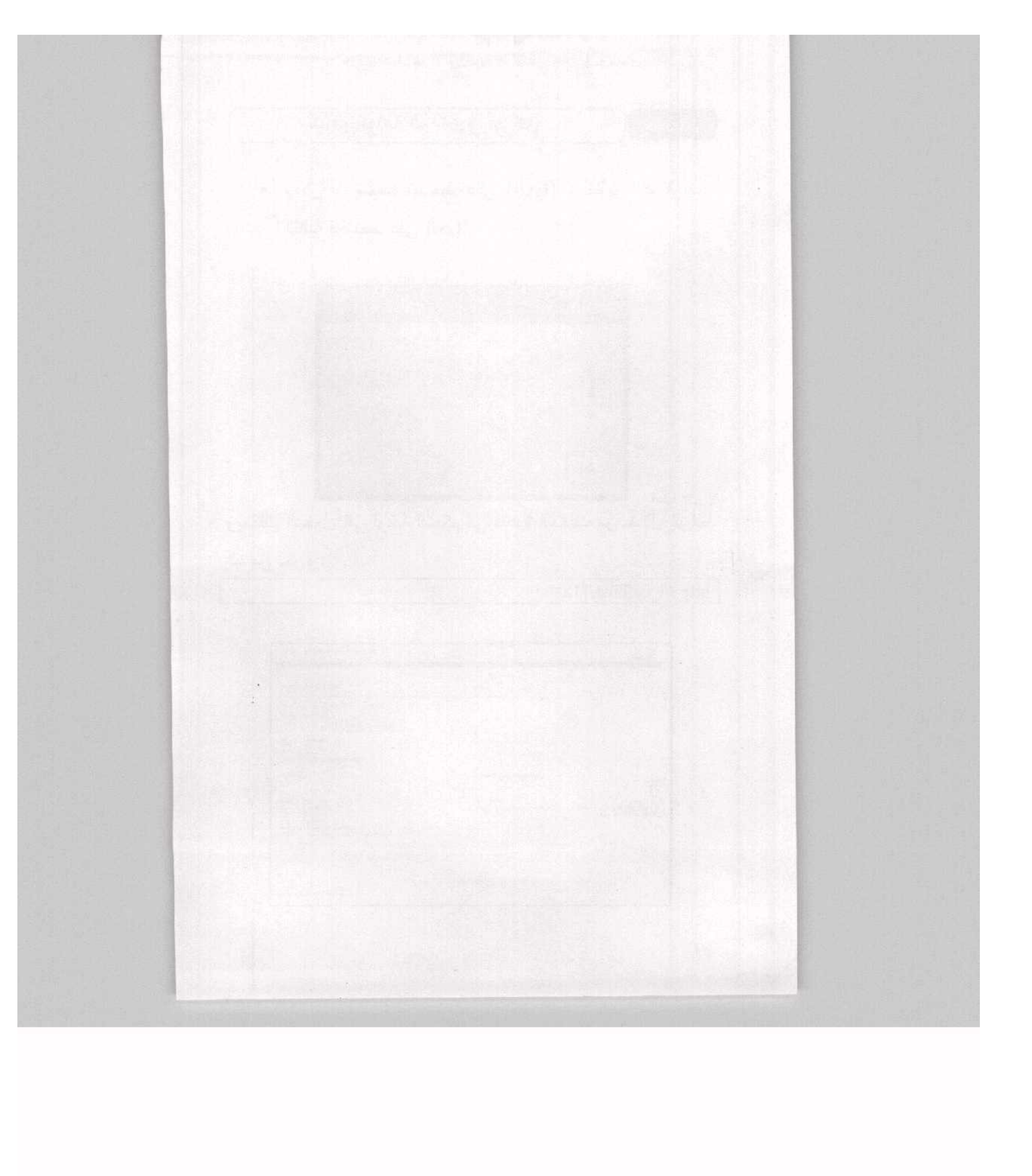
• والآن .. سوف تضغط على (Exit) ، لتظهر لك الرسالة التالية فاضغط على (نعم) :



ويمكنك الدخول إلى لوحة التحكم فى قاعدة البيانات من خلال الرابط السابق ذكره :

```
http://localhost:1158/em
```





الفصل الثانى

العمل فى بيئة iSQL * PLUS

Working In iSQL * PLUS

الـــ iSQL*PLUS لإدارة قاعدة البيانـــات ، مـــن خـــلال المتصفح ، وتعمل علي أي نظام . وهى تسمح بالدخول إلـــي نظـــام قاعدة البيانات بحسابات مختلفة ، كما تسمح بقراء بيانـــات الجـــداول والسجلات ، والقيام بالتعديل فيها .

ولنبدأ الآن فى العمل ..

لاشك أنك لاحظت أثناء تركيب البرنامج ، وعند نهايته تماما أنه ذكر لك عدة روابط يشبه التالى :

```
http://localhost:5561/isqlplus
```

وربما كان المنفذ (5560) ، أو غيره من المنافذ ، لكنه على أية حال يشبه السابق ، فالإختلاف ليس إلا فى رقم المنفذ .
وعند محاولة الدخول الى هذا الرابط سوف تظهر لك النافذة التالية :

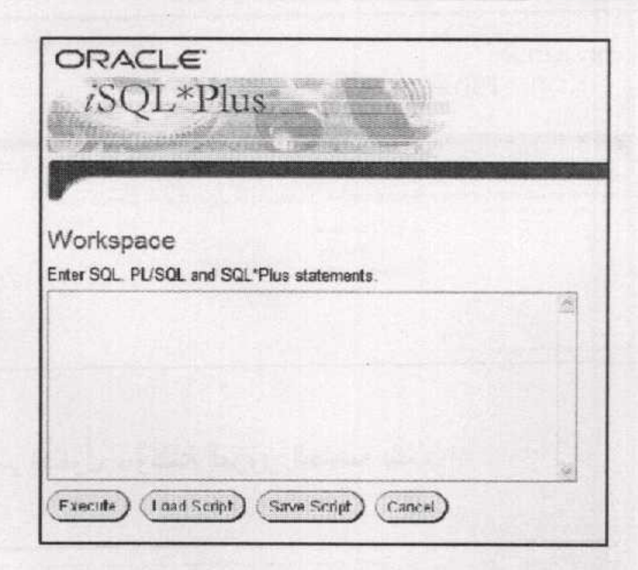
ORACLE
iSQL*Plus
Login
Unauthorized use of this site is prohibited and may be subject to civil and criminal prosecution.
* Indicates required field
* Username
* Password
Connect Identifier
Login
Help

اكتب اسم الدخول ، وكلمة المرور الخاصة بك :

* Username scott
* Password •••••
Connect Identifier
Login

لاحظ : أننا استعملنا المستخدم (Scott) للدخول ، ولم نستخدم المستخدم (System) ،وذلك لان قاعدة البيانات التي تعمل عليها تم إنشاؤها داخل الحساب (Scott).

هذا .. وإذا تمت عملية الدخول بنجاح يسظهر لك الشكل التالى :



ويمكنك كتابة أي عبارة استعلام – الآن – داخل صندوق النص (Work Space) . لكى تتعرف على النتيجة .

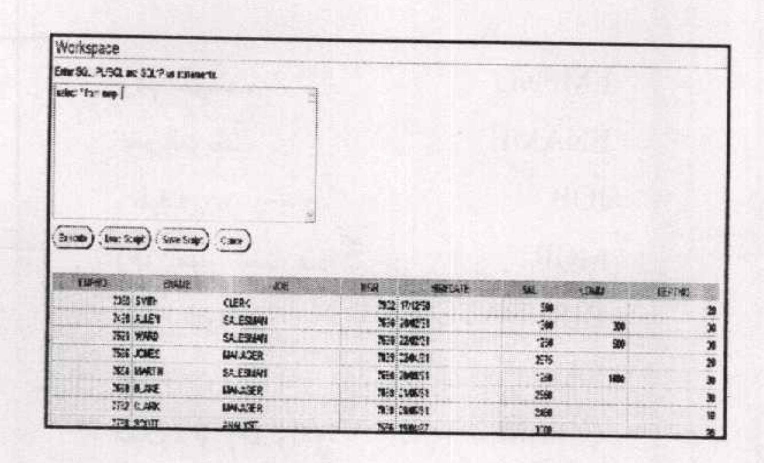
وفى السطور التالية سوف نتعرف على مجموعة جديدة من أوامر لغة الاستعلام ، وذلك خلال عدة أمثلة تقود فى النهاية إلي الطريق الصحيح ، فيما يخص هذا المجال .

المثال الأول :

هناك جدول باسم (emp) ، نريد الحصول على كافة البيانات المسجلة داخله. والوصول إلى ذلك يتم عبر الاستعلام التالى :

```
select * from emp;
```

وستكون النتيجة كالتالى :



حيث نجد فى الشكل السابق (Grid View) (أسفل الصفحة) ، يشير الى نتيجة تنفيذ الاستعلام ، ويظهر لك أيضا عدد السجلات الموجودة ، وهى 14 سجلا.

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17/12/80	800		20
7499	ALLEN	SALESMAN	7698	20/02/81	1600	300	30
7521	WARD	SALESMAN	7698	22/02/81	1250	500	30
7566	JONES	MANAGER	7839	02/04/81	2975		20
7654	MARTIN	SALESMAN	7698	28/09/81	1250	1400	30
7698	BLAKE	MANAGER	7839	01/05/81	2850		30
7782	CLARK	MANAGER	7839	09/06/81	2450		10
7788	SCOTT	ANALYST	7566	19/04/87	3000		20
7839	KING	PRESIDENT		17/11/81	5000		10
7844	TURNER	SALESMAN	7698	08/09/81	1500	0	30
7876	ADAMS	CLERK	7788	23/05/87	1100		20
7900	JAMES	CLERK	7698	03/12/81	950		30
7902	FORD	ANALYST	7566	03/12/81	3000		20
7934	MILLER	CLERK	7782	23/01/82	1300		10

14 rows selected

وكما ترى فإن الجول يتكونت من ثماني أعمدة ، هى من اليسار الى اليمين :

EMPNO	:	رقم الموظف
ENAME	:	اسم الموظف
JOB	:	وظيفته
MGR	:	رقم المدير المشرف
HIREDATE	:	تاريخ التوظيف
SAL	:	المرتب
COMM	:	العلاوة أو (العمولة)
DEPTNO	:	رقم القسم

المثال الثانى :

فى هذا المثال نود عرض كافة السجلات الموجودة داخل الجدول (emp) ، لكن فيما يتعلق بالأعمدة الخاصة برقم الموظف ، واسمه ، ووظيفته. وذلك يكون من خلال الاستعلام التالي :

```
select EMPNO,ENAME,JOB from emp ;
```

وستكون النتيجة كالتالى :

EMPNO	ENAME	JOB
7369	SMITH	CLERK
7499	ALLEN	SALESMAN
7521	WARD	SALESMAN
7566	JONES	MANAGER
7654	MARTIN	SALESMAN
7698	BLAKE	MANAGER
7782	CLARK	MANAGER
7788	SCOTT	ANALYST
7839	KING	PRESIDENT
7844	TURNER	SALESMAN
7876	ADAMS	CLERK
7900	JAMES	CLERK
7902	FORD	ANALYST
7934	MILLER	CLERK

14 rows selected.

وكما ترى فى الشكل السابق ، هناك ثلاثة أعمدة فحسب هى التى تم عرضها هى : (JOB) و (ENAME) و (EMPNO) ، وقد تم عرض 14 سجلا ، وهو مايعنى أن عدد الموظفين المسجلين بالشركة 14 موظفا .

المثال الثالث :

لاشك أنك لازلت تذكر أننا قنما في المثال الأول بعرض أرقام أقسام العمل الخاصة بالموظفين ، والتى ظهرت على يمين الشكل الخاص بذلك المثال على النحو التالى :

DEPTNO
20
30
30
20
30
30
10
20
10
30
20
30
20
10

ومن خلال الشكل السابق يمكنك أن تدرك أن هناك ثلاث أقسام يعمل بها الموظفين ، وهذا لايعنى أن الشركة تحتوي علي هذه الأقسام

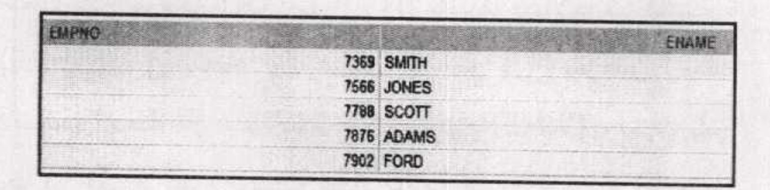
الثلاثة فقط فحسب ، فربما تكون هناك أقسام أخرى ، لكنها لايوجـد بها موظفون .
فالأقسام ثلاثة ؛ لأن هناك ثلاثة مجموعات من الأرقام هــى (10) و (20) و (30) . والموظفون موزعون عليها .

والآن .. نريد القيام باستعلام يوضح لنا الموظفين ، الذين يعملون فى القسم (20) .

إذن .. فسوف يكون الاستعلام على هذا المحو :

```
select EMPNO ,ENAME  from emp where DEPTNO =20 ;
```

وتكون النتيجة كما يلى :



EMPNO	ENAME
7369	SMITH
7566	JONES
7788	SCOTT
7876	ADAMS
7902	FORD

هذا .. ويمكنك التأكد من صحة الاستعلام (يدوياً) من خلال المثـال الأول .

المثال الرابع :

في المثال الأول عرضنا عمودا باسم (JOB) . وهو العمود الذى يصف وظيفة كل موظف داخل الشركة . فهناك من يعمل كـ (CLERK) وهناك من يعمل (MANAGER) .. وهكذا

وفى هذا المثال نريد اظهار الموظفين الذين يعملون كتبة (CLERKS) .

وذلك يتم من خلال الاستعلام التالى :

```
select EMPNO ,ENAME  from emp where JOB = 'CLERK' ;
```

وقد استخدمنا الشرط WHERE – Where Clause لتقتصر نتيجة البحث علي موظفي (CLERK) . وقد كتبناها بالأحرف الكبيرة (Capital) ، ووضعناها بين علامتى تنصيص (' ') مفردتين . وذلك لأننا حين نبحث عن سجلات تحت شرط (WHERE) ، بحيث يأتي بعد هذا الشرط عمود نستخدم عندئذ علامة اقتباس فردية (سينجل كوت) ، عندما نساويها بإحدى القيم .

وعلى ذلك ستكون نتيجة الاستعلام كم يلى :

EMPNO	ENAME
7369	SMITH
7876	ADAMS
7900	JAMES
7934	MILLER

المثال الخامس :

في هذا المثال – والذي يليه – سوف نبتعد قليلا عن جملة الاستعلام ، التي تظهر لنا بيانات السجل ، وسنستعرض جملتين تظهران بيانات الجدول وقاعدة البيانات .

من خلال الفصل السابق علمنا أن كل عمود له صفة بيانات ، سواء أكانت هذه البيانات نصية ، أو رقمية ، أو تاريخ ، أو عملة.
ولكننا لم ننشىء جدولا ، بل استخدمنا واحدا جاهزا لكى نتعلم من خلاله .

ولكى نتمكن من معرفة نوع البيانات المسموح بها في هذا الجدول لكل عامود ، فإن ذلك يتم من خلال جملة الاستعلام التالية :

```
desc emp;
```

لتكون نتيجة تنفيذ جملة الاستعلام كالتالى :

Name	Null?	Type
EMPNO	NOT NULL	NUMBER(4)
ENAME		VARCHAR2(10)
JOB		VARCHAR2(9)
MGR		NUMBER(4)
HIREDATE		DATE
SAL		NUMBER(7,2)
COMM		NUMBER(7,2)
DEPTNO		NUMBER(2)

وكما تري فى الشكل السابق فإن الجزء الشبكي مقسم إلى ثلاثة أجزاء هى :

Name : ويحتوي علي أسماء الأعمدة الموجودة في الجدول.
Null? : ويظهر ما إذا كان مسموحا ترك حقل العمود فارغاً أم لا.
Type : ويحدد نوع البيانات المسموح بها في حقل العمود وعددها .

إذن .. هناك ثمانية أعمدة :
الأول - EMPNO : وهو رقمي ، مسموح بكتابة أربعة أرقام داخله (NUMBER 4) ولا يسمح بتركه فارغاً .
الثاني - ENAME : وهو نصي ، يمكن كتابة 10 أحرف داخله .
الثالث - JOB : وهو نصي ، يمكن كتابة 9 أحرف داخله .

الرابع – MGR : وهو رقمي يمكن كتابة 4 أرقام داخله .

الخامس – HIREDATE : ويسمح بكتابة التاريخ فحسب .

السادس – SAL : وهو رقمي ، مخصص للعملة 7.2 .

السابع – COMM : وهو رقمي ، مخصص أيضا للعملة 7.2 .

الثامن – DEPTNO : وهو رقمي ، يسمح بكتابة رقمين فحسب .

وعلى هذا النحو نكون قد تعرفنا على الأعمدة الموجودة فى الجدول (emp) ، كما عرفنا نوع بياناتها.

المثال السادس :

تحتوي قاعدة البيانات علي عدة جداول ، فالجدول (emp) واحد من عدة جداول داخل المستخدم (Scott) . وقد رأينا في المثال الأول أنه لابد من معرفة اسم الجدول لكى يمكن تنفيذ جملة الاستعلام .

أما كيفية معرفة الجداول الموجودة داخل قاعدة البيانات لكى تتم كتابة جملة استعلام صحيحة فهى تتم من خلال تنفيذ الاستعلام التالي :

```
select * from tab ;
```

وستكون نتيجة تنفيذ الاستعلام كالتالى :

TNAME	TABTYPE
DEPT	TABLE
EMP	TABLE
BONUS	TABLE
SALGRADE	TABLE

وكما تري في الشكل السابق ، هناك أربعة جداول في قاعدة البيانات هى :

- DEPT : وهو الخاص بالأقسام التي يعمل الموظفون بها.
- EMP : وهو خاص بالموظفين.
- Bonus : وهو جدول فارغ .
- SALGRADE : وهو خاص بالمرتب .

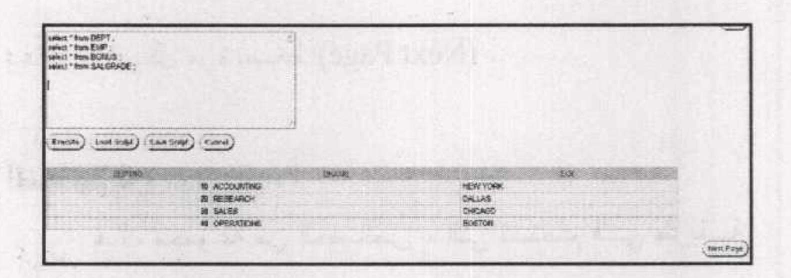
المثال السابع :

فى هذا المثال نحاول الاستعلام عن كافة البيانات الموجـودة داخــل الجدول (والتي رأيناها في المثال السابق) .

ويمكن تنفيذ ذلك من خلال كتابة مجموعة من جمل الاستعلام التاليــة فى مكان العمل (Work Space) :

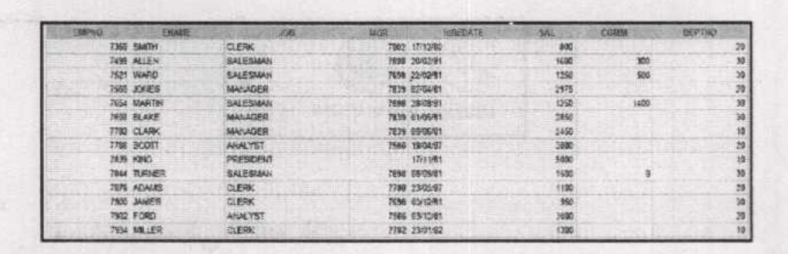
```
select * from DEPT ;
select * from EMP ;
select * from BONUS ;
select * from SALGRADE ;
```

عندئذ ستظهر لك بيانات الجدول DEPT :



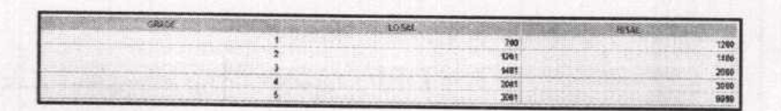
DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

- اضغط علي الزر (Next_Page) ، وسيظهر لــك الجــدول (EMPNO) :



EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17/12/80	800		20
7499	ALLEN	SALESMAN	7698	20/02/81	1600	300	30
7521	WARD	SALESMAN	7698	22/02/81	1250	500	30
7566	JONES	MANAGER	7839	02/04/81	2975		20
7654	MARTIN	SALESMAN	7698	28/09/81	1250	1400	30
7698	BLAKE	MANAGER	7839	01/05/81	2850		30
7782	CLARK	MANAGER	7839	09/06/81	2450		10
7788	SCOTT	ANALYST	7566	19/04/87	3000		20
7839	KING	PRESIDENT		17/11/81	5000		10
7844	TURNER	SALESMAN	7698	08/09/81	1500	0	30
7876	ADAMS	CLERK	7788	23/05/87	1100		20
7900	JAMES	CLERK	7698	03/12/81	950		30
7902	FORD	ANALYST	7566	03/12/81	3000		20
7934	MILLER	CLERK	7782	23/01/82	1300		10

- اضغط علي الزر (Next Page)،وســيظهر لــك الجــدول (SALGRADE) :



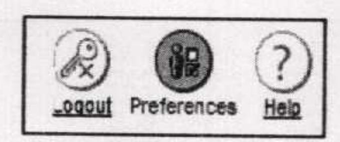
GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

وهكذا .. فى كل مرة نضغط (Next Page) .

التفضيلات Preferences

هناك مجموعة من الخصائص ، التي تـتحكم فــي طريقــة عرض البيانات فى صفحة العمل .

- ولضبط هذه الخصائص اضغط علي الرابط (preferences) الموجود في أعلي الصفحة :



- ثم توجه إلي النافذة التالية :

• Interface Configuration
• System Configuration
  ○ Script Formatting
  ○ Script Execution
  ○ Database Administration
• Change Password

Interface Configuration
Configure settings that affect the iSQL*Plus user interface.

History Size
Set the number of scripts displayed in the script history.
Scripts 10

Input Area Size
Set the size of the script input area.
Width 70
Height 10

Output Location
Set where script output is displayed.
◉ Below Input Area
○ Save to HTML File

Output Page Setup
Set whether output is dislayed on a single page, or over multiple pages.
○ Single page
◉ Multiple pages
Number of rows on each page 24
Text at each page break

في المثال (7) السابق كانت نتائج الاستعلام تظهر بالضغط على الزر (Next Page) . ولكننا نريد نريد أن تظهر كافة النتائج في صـــفحة واحدة .

• إذن .. توجه إلي القسم OUTput Page Setup ، وستجده مضبوطا على عرض 24 سجلا ، في الصفحة الواحدة ، كما أنه مضبوط علي Multi pages :

Output Page Setup

Set whether output is dislayed on a single page, or over multiple pages.

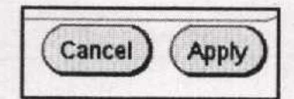
○ Single page

◉ Multiple pages

Number of rows on each page [24]

Text at each page break [ ]

- لا نغير شيئا فى الرقم ، بل اختر Single page ، ثم اضغط Apply .



- توجه إلي أسفل الصفحة واضغط علـــي Workspace ، أو توجه إلى أعلي الصفحة واضغط علي اللسان Workspace، وسوف تجد أنك فى صفحة تنفيذ جمل الاستعلام :

Workspace | History | Logout | **Preferences** | Help

قم بتنفيذ جملة الاستعلام السابقة مع المثال السابع ، وسوف تجد نتيجة ما قمنا به من عمل فى صفحة التفضيلات :

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17/12/80	800		20
7499	ALLEN	SALESMAN	7698	20/02/81	1600	300	30
7521	WARD	SALESMAN	7698	22/02/81	1250	500	30
7566	JONES	MANAGER	7839	02/04/81	2975		20
7654	MARTIN	SALESMAN	7698	28/09/81	1250	1400	30
7698	BLAKE	MANAGER	7839	01/05/81	2850		30
7782	CLARK	MANAGER	7839	09/06/81	2450		10
7788	SCOTT	ANALYST	7566	19/04/87	3000		20
7839	KING	PRESIDENT		17/11/81	5000		10
7844	TURNER	SALESMAN	7698	08/09/81	1500	0	30
7876	ADAMS	CLERK	7788	23/05/87	1100		20
7900	JAMES	CLERK	7698	03/12/81	950		30
7902	FORD	ANALYST	7566	03/12/81	3000		20
7934	MILLER	CLERK	7782	23/01/82	1300		10

14 rows selected.

no rows selected.

GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

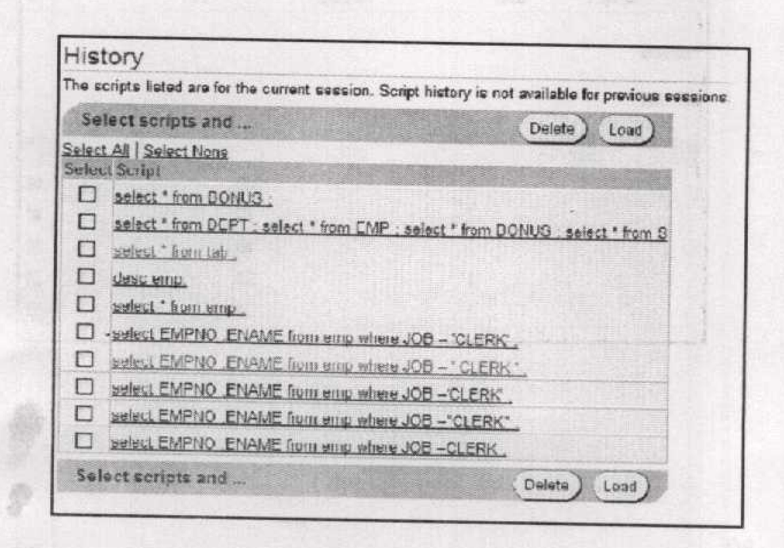
الذاكرة History

حين ننفذ الصفحة (History) يتم حفظ جمل الاستعلام في ذاكرة الأداة iSQL*Plus ، بحيث يمكننا الرجوع إلى الجملة مرة أخرى دون الحاجة إلى كتابتها مرة أخرى .

ويمكنك التوجه إلي الذاكرة بالضغط علي اللسان (History) ، الموجود في أعلي الصفحة :



وسوف تظهر لك النافذة التالية :



وكما هو واضح من الشكل فقد تم حفظ جمل الاستعلام العشر الأخيرة ، التى سبق أن قمت بتنفيذها . وبناء عليه يمكنك الآن الضغط على أى جملة منها لاستخدامها مرة أخرى دون الحاجة إلي كتابتها .

ولكننا نريد أن تحتفظ ذاكرة الأداة iSQL*Plus بعدد أكبر من جمل الاستعلام ، التي تستخدمها .

- توجه إلي صفحة التفضيلات ، وستجد فى المنطقة (History Size) أن الرقم (10) موجود في الخانة (Scripts) :

History Size

Set the number of scripts displayed in the script history.

Scripts | 10

- ثم قم بكتابة (100) بدلاً من (10) :

Scripts | 100

- ثم اضغط علي الزر (Apply) ليتم حفظ التغييرات ، التي قمت بها :



المثال الثامن :

تعرفنا في المثال الرابع على كيفية البحث عن الموظفين الذين يعملون (CLERK) . وكان ذلك من خلال جملة الاستعلام التالية :

```
select EMPNO ,ENAME  from emp where JOB = 'CLERK' ;
```

وقد وضعنا كلمة (CLERK) بين علامتي تنصيص (إقتبـــاس (' ')) فرديتين ، والسبب أننا حين نبحث عن سجلات بشرط (WHERE) ، يأتي بعدها عمود نصي ، لذا نستخدم علامة (اقتباس) فردية (سينجل كوت) عند مساواتها بإحدى القيم ، كما سبق بيان ذلك فـــى صـــفحة سابقة .

إلا أننا فى الأعمدة الرقمية لا نستخدم أى علامات اقتباس .

والآن .. نريد البحث عن الموظفين الذين يعملون براتب 1000 جنيه، وذلك كمثال للبحث عن حقل رقمي بعد شرط (Where) .
وذلك ممكن من خلال تنفيذ الاستعلام التالي :

```
select EMPNO,ENAME,JOB from emp where sal >=1000;
```

حيث ستكون النتيجة كالتالى :

EMPRO	ENAME	JOB
7499	ALLEN	SALESMAN
7521	WARD	SALESMAN
7566	JONES	MANAGER
7654	MARTIN	SALESMAN
7698	BLAKE	MANAGER
7782	CLARK	MANAGER
7788	SCOTT	ANALYST
7839	KING	PRESIDENT
7844	TURNER	SALESMAN
7876	ADAMS	CLERK
7902	FORD	ANALYST
7934	MILLER	CLERK

12 rows selected.

المثال التاسع :

لو أننا أردنا البحث عن موظف معين بالاسم ، وليكن (MARTIN)، بحيث نستعرض رقمه ووظيفته ، فإن ذلك يتم من خلال تنفيذ جملة الاستعلام التالية ، في مكان العمل :

```
select EMPNO,JOB from emp where ENAME = 'MARTIN' ;
```

وقد وضعنا اسم الموظف بين علامتي (اقتباس) فرديتين . ولكن قـد يختلط علينا هجاء الاسم فلا نعرف مـا إذا كـان (MARTEN) أو (MRTEN) ، عند ذلك يمكننا البحث من خلال الحرف الأول فـي الاسم ، وذلك من خلال تنفيذ جملة الاستعلام التالية :

```
select EMPNO,ENAME,JOB from emp where ENAME like
'm%' order by ENAME;
```

وتجب ملاحظة أننا كتبنا حرف (m) صـغير (small) ثـم تبعنـاه بعلامة النسبة المئوية (%) وذلك بالبحث عن أى اسم لأحد الموظفين يبدأ بحرف (m).

وستكون نتيجة تنفيذ الاستعلام كالتالى :

```
select EMPNO,ENAME,JOB from emp where ENAME like 'm%' order
by ENAME;

Execute   Load Script   Save Script   Cancel
no rows selected
```

وقد كانت النتيجة – كما هو واضح مـــن الـــشكل الـــسابق – هـــى (norows selected) ، وهذا يعني عدم وجود موظفين تبدأ أسماؤهم بالحرف (m) .

وهو أمر غير ، إذ أننا نعرف أن هناك موظفين يبدأ اسمهم بحـــرف (m) ، فأين الخطأ ؟!

إن الخطأ يكمن فى أن الموظفين المسجلين فى الجدول وتبدأ أسماؤهم بحرف (m) تم تسجيل اسماؤهم بالحروف الكبيرة (كابيتال) ، لذا فقد كانت طريقة البحث خاطئة .

لذا .. يجب تنفيذ جملة الاستعلام بحيث يكون الحـــرف (m) كبيـــرا (كابيتال M) . وعلى ذلك سوف تكون جملة الاستعلام كما يلى :

```
select EMPNO,ENAME,JOB from emp where ENAME like
'M%' order by ENAME;
```

عندئذ ستكون النتيجة كالتالي :

EMPNO	ENAME	JOB
7654	MARTIN	SALESMAN
793[illegible]	MILLER	CLERK

المثال العاشر :

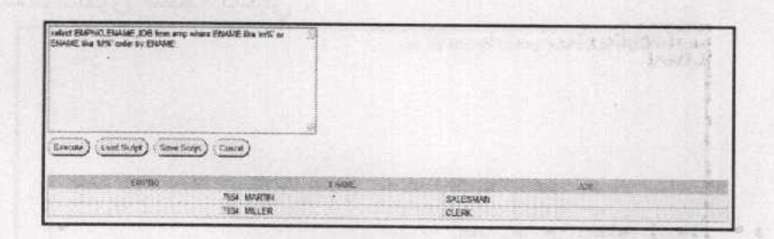
عرفنا فى كيف يمكن البحث عن موظفين يبدأ اسمهم بحرف صـــغير (سمول) . ثم عرفنا كيف نبحث عمن يبــدأ اســـمهم بحـــرف كبيـــر (كابيتال). لكننا لم نعرف كيف نبحث بطريقة أفضل .

لذلك فسوف نكتب جملة استعلام نبحث فيها عن حرفين ، سواء أكان أحدهما (سمول) والآخر (كابيتال) ، أو كانا مختلفين تماما مثل (S) و (D):

ويمكن القيام بهذا من خلال جملة الاستعلام التالية :

```
select EMPNO,ENAME,JOB from emp where ENAME like
'm%' or ENAME like 'M%' order by ENAME;
```

وستكون النتيجة كما يلى :



المثال الحادى عشر :

عرفنا في الأمثلة السابقة كيف يمكننا البحث عن سجلات بشرط ، تأتي بعده عملية مساواة (=) ، لكننا لم نعرف بعد كيف يمكن حصر السجلات بين (متغيرين) ، لا متغير واحد .

فإذا أردنا القيام بعرض سجلات الموظفين ، الذين يتراوح راتب الواحد منهم بين عددين ، مثل (2000 جنيه) و (3000 جنيه) فإن ذلك يمكن أن يتم من خلال استخدام المشغلات (Operators) ، مثل (=>) و (=<) .

وهذا يتحقق من خلال كتابة جملة الاستعلام التالية في مكان العمل :

```
select EMPNO,ENAME,SAL from emp where SAL >= 2000
and SAL <=3000;
```

وستكون النتيجة كالتالى :

EMPNO	ENAME	SAL
7566	JONES	2975
7698	BLAKE	2850
7782	CLARK	2450
7788	SCOTT	3000
7902	FORD	3000

المثال الثانى عشر :

فى هذا المثال سنحاول البحث عن القيم الفارغة (Null). حيــث ان هناك من الحقول داخل الأعمدة والسجلات ماهو ملئ بالبيانات ، بينما يوجد غيره فارغا .

لذلك سوف نستعرض كيفية البحث عن الحقل الفارغ ..

فى المثال الأول كان العمود (COMM) يحتوي علي بعض الحقول التى بها بيانات ، كما كان يحتوى على حقول أخرى فارغة .

فإذا أردنا – على سبيل المثــال – البحــث عــن المــوظفين الــذين لايحصلون على علاوة ، بمعنى هؤلاء الذين تكون خانة (COMM) الخاطئة بهم فارغة (NULL) ..

فإن هذا يمكن أن يتم من خلال تنفيذ جملة الاستعلام التالية :

```
select EMPNO,ENAME,COMM  from emp where COMM is
null;
```

وسوف تجد النتيجة كالتالى :

EMPNO	ENAME	COMM
7369	SMITH	
7566	JONES	
7698	BLAKE	
7782	CLARK	
7788	SCOTT	
7839	KING	
7876	ADAMS	
7900	JAMES	
7902	FORD	
7934	MILLER	

10 rows selected

المثال الثالث عشر :

بحثنا في المثال السابق عن قيمة فارغة (NULL) . وفى هذا المثـــل نستعرض كيفية البحث عن قيمة غير فارغة .

ولاشك أن البحث عن قيمة فارغة أمر مفهوم ، لكن غير المفهوم هو البحث عن قيمة غير فارغة ، إذ أن هذه القيمة تظهر بشكل طبيعـــى عند تنفيذ أى استعلام .

لكننا قد نود عمل حصر للموظفين الذين يحصلون على عـــلاوة ، أى الذين لهم بيانات مكتوبة فى العمود (COMM) .

وهذا يمكن تنفيذه . من خلال كتابة جملة الاستعلام الثالثة فى مكـــان العمل .

```
select EMPNO,ENAME,COMM  from emp where COMM is not null;
```

لنكون نتيجة تنفيذ الاستعلام كما يلى :

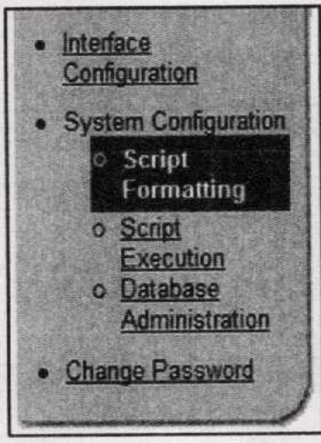
EMPNO	ENAME	COMM
[illegible]	ALLEN	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	MARTIN	[illegible]
[illegible]	TURNER	[illegible]

إظهار عدد السجلات :

إذا عدت إلى الأمثلة السابقة ستجد ان الجزء الشبكى لايظهـــر عـــدد السجلات أسفله ، والسبب فى ذلك أن عدد السجلات يكون أقل مـــن (6) ، وهو العدد الذى تم ضبط الأداة iSQL*Plus عليه .

وبطبيعة الحال ، فمن الممكن تغغير هذا الوضع ، وذلك على النحـــو التالى ..

توجه إلي صفحة التفضيلات ، ثم اضغط علي Script Formatting الموجودة على يسار الصفحة :



وسوف تظهر لك صفحة أخرى ، يوجد بها القسم Display Record Count حيث ستجد الرقم (6) موجودا فـــي الخانـــة Number of records :

Display Record Count
Set whether to display the number of records returned by queries or PL/SQL blocks.
◉ On
○ Off
Number of records 6

فقم بكتابة الرقم (1) بدلاً من (6) ، ثم اضغط (Apply):

Number of records 1

فإذا قمت بتنفيذ جملة الاستعلام ، التى فى هذا المثال أو غيره ســوف يظهر لك عدد السجلات :

EMPNO
7499
7521
7654
7844

4 rows selected.

المثال الرابع عشر :

لو عدنا الى جدول الموظفين (emp) ، فـسنجد عمـودين للعملـة : أولهما للراتب ، والثانى للعمولة (أو العلاوة) . وذلك لأن الموظـف يحصل على الراتب بشكل أساسى ، كما قد يحصل على عمولة ، أو علاوة .

وفى هذا المثال نريد عرض المبلغ الإجمالى الذى يتسلمه الموظفون ، شاملا الراتب والعلاوة ، وهذا معناه أننا سنقوم بجمع الحقلين (SAL) و (COMM) الخاصين بكل موظف ، ثم نعرض الناتج فى الجـزء الشبكي .

ويمكن فعل ذلك من خلال كتابة جملة الاستعلام التالية :

```
select EMPNO,ENAME,SAL+COMM  from emp ;
```

لتجد أن نتيجة تنفيذ الاستعلام كما يلى :

EMPNO	ENAME	SAL+COMM
7369	SMITH	
7499	ALLEN	1900
7521	WARD	1750
7566	JONES	
7654	MARTIN	2650
7698	BLAKE	
7782	CLARK	
7788	SCOTT	
7839	KING	
7844	TURNER	1500
7876	ADAMS	
7900	JAMES	
7902	FORD	
7934	MILLER	

14 rows selected.

هذا .. ويمكنك تنفيذ الاستعلام السابق ، مع عرض حقلي الراتب والعلاوة ، من خلال تنفيذ الاستعلام التالي :

```
select ENAME,SAL,COMM,SAL+COMM  from emp ;
```

وستكوم النتيجة كالتالى :

ENAME	SAL	COMM	SAL+COMM
SMITH	800		
ALLEN	1600	300	1900
WARD	1250	500	1750
JONES	2975		
MARTIN	1250	1400	2650
BLAKE	2850		
CLARK	2450		
SCOTT	3000		
KING	5000		
TURNER	1500	0	1500
ADAMS	1100		
JAMES	950		
FORD	3000		
MILLER	1300		

14 rows selected.

واذا نظرت الى الجزء الشبكى ، ستجد أن العمود SAL+COMM يحتوى على حقول بيانات فارغة (NULL) . والسبب فى ذلك أن الموظفين الذين تخصهم هذه الحقول لا يحصلون علي علاوة . ولكنك قد تتساءل : إذن فلماذا لم تتم كتابة الراتب فحسب ؟

والإجابة عن سؤالك هذا نقول : إن أى قيمة يتم جمعها على (NULL) سوف تساوي (NULL) وبمعني أكثر تبسيطا . لو أن

أحد الموظفين يحصل على 1000 جنيه كراتــب ، دون أن يحــصل على علاوة ففى حالة جمع الحقلين معا سيتم جمع (1000 جنيــه) + NULL ، وهناك تكون النتيجة (NULL) وليست الألف جنيه .

ولو دققت النظر من جديد فــي الــشكل الــسابق ســتجد أن الموظف (Turner) هو الوحيد الذي لا يحصل على علاوة ، ومــع ذلك ظهر راتبه في حقل (SAL+COMM) ، والسبب فى ذلك أن القيمة المكتوبة في الحقل (COMM) الخاص بــه هــي (صــفر) ، وليست (NULL) .

والقيمة (NULL) لا تعني (صفرا) ، وإلا لم تكن هنــاك مــشكلة ، لكنها تعنى (فارغ) وهناك فرق بين (صفر) و (فارغ) .

ولحل هذه المشكلة سنعرض طريقتين لذلك ، مــن خــلال المثــالين التاليين .

المثال الخامس عشر :

في المثال السابق عرضنا المبلغ الإجمــالى ، الــذى يحــصل عليــه الموظف ، متضمنا الراتب والعلاوة ، أى أننــا جمعنــا الحقلــين : (SAL) و (COMM) لكل موظف . ولكننــا وجــدنا أن العمــود (SAL+COMM) يحتوي علي حقول بيانات فارغــة (NULL) ، والسبب أن الموظفين ، الذين تخصهم هذه الحقول لا يحصلون علــي

علاوة . ومع ذلك كان من المفترض أن يكون المكتوب هو الراتـــب وحده ، ولكن الأمر جاء على خلاف ذلك لأن أى قيمة تجمـــع علـــي (NULL) فسوف تساوي (NULL) .

فلو أن أحد الموظفين يحصل على راتب 1000 جنيه ، ولا يحـــصل علي علاوة، فإن حاصل جمع الحقلـــين ســـيكون 1000+ NULL ، فتكون النتيجة (NULL) وليس ألف جنيه .

ويمكن حل المشكلة من خلال عرض ســـجلات المـــوظفين ، الـــذين يصرفون علاوات ، لا كل الموظفين .

ويمكن تنفيذ ذلك من خلال جملة الاستعلام التالية :

```
select ENAME,SAL,COMM,SAL+COMM   from emp where
COMM is not null ;
```

وتكون النتيجة كما يلى :

ENAME	SAL	COMM	SAL+COMM
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]

[illegible]

المثال السادس عشر :

في المثال (الرابع عشر) عرضنا المبلغ الكلى الـذى يحـصل عليـه الموظفون ، متضمنا الراتب والعلاوة ، أى قمنا بجمع حقلى (SAL) و (COMM) . لكل موظف .

وفي المثال (الخامس عشر) قمنا بحـل المـشكلة الخاصـة بجمـع (NULL) مع قيمة ، وذلك من خـلال عـرض المـوظفين الـذين يحصلون علي علاوة فحسب ، وتنفيذ عملية الجمـع علـيهم ، دون غيرهم .

ولكن هذا الحل ليس عمليا !

فنحن نرغب فى عرض الجميع ، مع المبالغ التى يحصلون عليهـا ، وفي حالة عدم حصول أحدهم على علاوة يتم عرض مرتبه .

وذلك ممكن من خلال التعويض عن القيمة (NULL) بصفر .

وهذا يتم إذا قمنا بتنفيذ جملة الاستعلام التالية :

```
select ENAME,SAL,COMM,SAL+nvl(COMM,0)  from emp
;
```

وفي هذه الجملة استخدمنا الشرط (nvl) ، حيث أن وظيفتها باستبدال القيمة التى نريدها بها ، وليس القيمة صفر .

فإذا أردنا استبدال الرقم 100 بالقيمة (100) ، يكون الأمر على النحو التالي :

```
select  ENAME,SAL,COMM,SAL+nvl(COMM,100)      from
emp  ;
```

وتكون نتيجة الاستعلام كما يلي :

ENAME	SAL	COMM	SAL+NVL(COMM,100)
SMITH	800		900
ALLEN	1600	300	1900
WARD	1250	500	1750
JONES	2975		3075
MARTIN	1250	1400	2650
BLAKE	2850		2950
CLARK	2450		2550
SCOTT	3000		3100
KING	5000		5100
TURNER	1500	0	1500
ADAMS	1100		1200
JAMES	950		1050
FORD	3000		3100
MILLER	1300		1400

14 rows selected.

المثال السابع عشر :

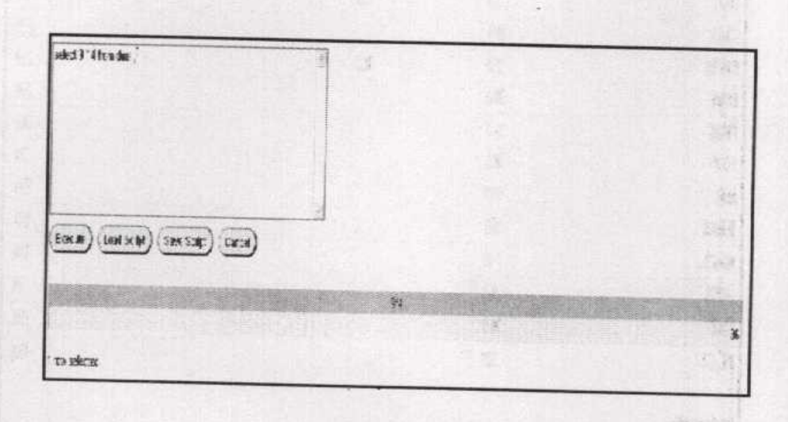
فى هذا المثال سننتقل إلى مجموعة جمل أخرى تقوم بوظائف جديدة .

فلو أننا أردنا معرفة حاصل ضرب رقمين مثل 9 × 4 .. فإن ذلـك من الممكن عمله باستخدام الأداة iSQL*PLUS ؟؟ ..

ويمكن تنفيذ ذلك من خلال جملة الاستعلام التالية :

```
select 9 * 4 from dual ;
```

وتكون هذه هى نتيجة الاستعلام :



لاحظ : أن الجدول (Dual) جدول وهمي غير موجود فى واقع الأمر ، ولكن يمكننا تنفيذ مجموعة من الاستعلامات الخارجية بواسطته .

ولكى نتأكد من ذلك اكتب الاستعلام التالي :

```
select * from dual ;
```

وستكون النتيجة كالتالى :

DUM
X

1 row selected.

كما يمكنك أن تكتب الاستعلام التالى فى الجزء الشبكى :

```
select 82753185 from dual ;
```

فتكون النتيجة كالتالى :

[illegible]
[illegible]

[illegible]

مع ملاحظة أنه لايمكنك تنفيذ الاستعلام السابق باستخدام حرف بـــدلا من الرقم فمثلا ، الاستعلام التالى :

```
select e from dual ;
```

سوف ينتج عنه رسالة الخطأ التالية :

```
select e from dual
       *

ERROR at line 1:
ORA-00904: "E": invalid identifier
```

يمكنك – كذلك – معرفة باقي قسمة رقمين ، من خـــلال الاســـتعلام التالي :

```
select mod(35,3) from dual ;
```

وهذه هى نتيجة تنفيذ الاستعلام :

MOD(35,3)
2

هذا .. ويمكنك معرفة تاريخ اليوم علـــي (الخـــادم) المركـــب عليـــه أوراكل ، من خلال الاستعلام التالي :

```
select SYSDATE from dual ;
```

الذى تكون نتيجته كالتالى :

SYSDATE
24/03/07

1 row selected.

كذلك يمكنك معرفة حاصل قسمة رقمين بكتابة الاستعلام التالي :

```
select 9/4 from dual ;
```

ونتيجة الاستعلام كما يلى :

9/4
2.25

ويمكنك كتابة ما تريد أن يظهر قبـــل كلمـــة (from) ، أي ســـيكون الاستعلام كالتالي :

```
select 9/4 Result from dual ;
```

وتكون النتيجة هكذا :

RESULT
2.25

ويمكنك معرفة حاصل قسمة رقمين (بنتيجة رقم صحيح) إذا كتبـــت الاستعلام التالي :

```
select trunc(9/4) from dual ;
```

وهذه هي النتيجة :

TRUNC(9/4)
2

كما يمكنك – أيضا – معرفة حاصل قسمة رقمين (بنتيجة رقم مقرب لأقرب عشرة) من خلال الاستعلام التالي :

```
select round(9/4,1) from dual ;
```

وهذه هى النتيجة :

ROUND(9/4,1)
2.3

وتستطيع معرفة عدد أحرف كلمة بعينها ، من خلال جملة الاستعلام التالية في مكان العمل :

```
select length ('alexandria') from dual;
```

وستجد النتيجة هكذا :

LENGTH('ALEXANDRIA')
10

المثال الثامن عشر :

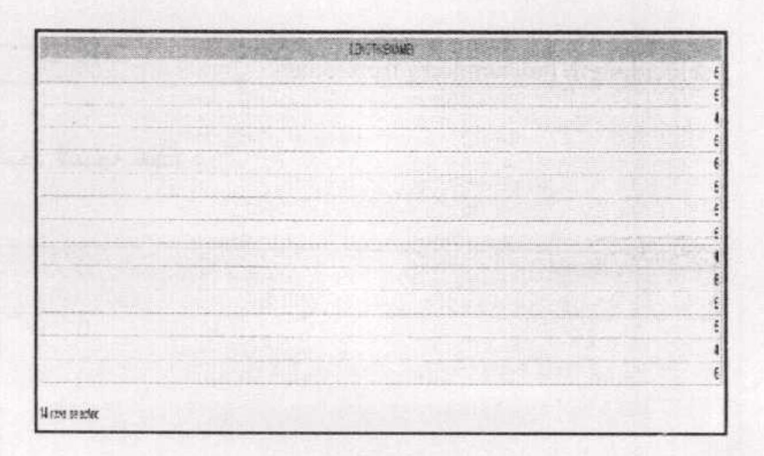
كانت آخر فقرة فى المثال السابق هى كيفية معرفة عدد أحرف كلمة بعينها ، من خلال (Length) بالاستعلام التالي :

```
select length ('alexandria') from dual;
```

الآن نريد معرفة عدد الحروف لكل موظفي الشركة في الجدول (emp) .. ويمكننا ذلك من خلال تنفيذ جملة الاستعلام التالية :

```
select length (ENAME) from emp;
```

ونتيجة لذلك سوف ينتج الآتى :



LENGTH(ENAME)
5
5
4
5
6
5
5
5
4
6
5
5
4
6

14 rows selected.

المثال التاسع عشر :

في معظم الأمثلة السابقة استعرضنا كيفية عــرض بيانــات لجــدول موجود بالفعل من قبل . لكننا لم نبين كيف يمكن إنشاء جدول جديد ، من خلال data definition language .

إذن .. سنقوم بإنشاء جدول باسم (address) يحتوي علــي حقلــين الأول : رقم الموظف ، والثاني : عنوانه.

وسيكون الحقل الأول رقميا والثاني نصيا .

ولإنشاء الجدول الذى نريده يمكننا استخدام جملة تعريــف البيانــات التالية :

```
create table address (EMPNO number(4),ADDRESS
varchar2(100)) ;
```

لتكون نتيجة تنفيذ جملة التعريف كالآتى :

```
Workspace
Enter SQL, PL/SQL and SQL*Plus statements.
create table address (EMPNO number(4),ADDRESS varchar2(100)) ;

Execute   Load Script   Save Script   Cancel
Table created
```

حيث تلاحظ الرسالة (Table Created) ، وهو مايعنى أنك نجحت فى إنشاء الجدول المطلوب .

فإذا حاولت عرض البيانات المسجلة في الجدول (address) من خلال كتابة جملة الاستعلام التالية :

```
select * from address;
```

فستكون النتيجة عدم ظهور أية سجلات

```
select * from address;
```

Execute | Load Script | Save Script | Cancel

no rows selected

والسبب .. أنك لم تقم بإدخال أي بيانات إلي الجدول حتى الآن .

المثال العشرون :

عندما قمنا بإنشاء جدول في المثال السابق لم نجد أى بيان حين قمنا بتنفيذ جملة استعلام لعرض البيانات ، وذلك لأننا لم ندخل أي بيان .

والآن .. سوف نقوم بإدخال بيانيين لاثنين من الموظفين ، وهو مايمكن عمله من خلال تنيذ جملتى معالجة متتاليتين : الأولى :

```
insert into address values (7369,'Egypt-Alexandria-15-Myami');
```

وبهكذا الشكل نكون أدخلنا الموظف رقم 7369 ، وهو موظف مسجل بجدول الموظفين (emp) ، وعنوانه :

```
Egypt-Alexandria-15-Myami
```

والجملة الثانية :

```
insert into address values (7499,'Egypt-Alexandria-16-Asafra');
```

وبهذا نكون قد أدخلنا البيان الخاص بالموظف رقم 7499 ، المسجل بجدول الموظفين ، وعنوانه : 'Egypt-Alexandria-16-Asafra .

وستكون نتيجة تنفيذ كل منهما مشابهة للنتيجة التالية :

```
insert into address values (7369,'Egypt-Alexandria-15-Myami');

Execute   Load Script   Save Script   Cancel
1 row created.
```

ويمكنك تأكيد أي تغييرات باستخدام الأمر التالي :

```
commit;
```

لتكون النتيجة هكذا :

```
commit;

Execute   Load Script   Save Script   Cancel
Commit complete.
```

ثم قم بتنفيذ جملة الاستعلام التالية :

```
select * from address;
```

لتكون النتيجة كالتالى :

EMPNO	ADDRESS
[illegible]	Egypt-Alexandria-15 Miami
[illegible]	Egypt-Alexandria-16 Asafra

المثال الحادى والعشرين :

إذا أردت معرفة موقع بداية كلمة ، فإن ذلك من خلال الدالة (INSTR) ، فهى التى تقوم بالبحث عن الكلمة وتحديد موقعها في الحقل .

فإذا أردنا أن نعرف موقع كلمة (Asafra) في الحقل (address) في سجلات الجدول ، فإن ذلك ممكن من خلال تنفيذ الأمر التالي :

```
select instr (ADDRESS,'Asafra') from address;
```

وتكون نتيجة تنفيذ الاستعلام كالتالى :

INSTR(ADDRESS,'ASAFRA')
0
21

وكما تري .. فإن السجل الأول أعطي القيمة (صفر) ، وهذا يدل علي عدم وجود الكلمة في السجل الأول .بينما السجل الثاني أعطى الـــرقم (21) ، وهو ما يدل على أن الكلمة تبدأ من هذا الموقع .

والآن .. سوف نحاول معرفة موقع أول العلامات في الحقل باستخدام الدالة ذاتها .

```
select instr (ADDRESS,'-') from address;
```

لتكون النتيجة كالتالى :

INSTR(ADDRESS,'-')
6
6

وكما ترى .. فإن الحقل يحتوي علي أكثر من علامة ، فإذا كنـــت لا تريد معرفة العلامة الأولي ، بل تريد الثانية . فيمكنك عمل ذلك مـــن خلال تنفيذ جملة الاستعلام التالية :

```
select instr (ADDRESS,'-',1,2) from address;
```

فالرقم (1) يدل علي مكان بداية الحقل ، بمعنى أننا سوف نبدأ البحث بداية من أول حرف .

والرقم (2) يعني البحث عن العلامة الثانية ، بمعنى أن البحث سيكون عنها ، وحين يجدها أول مرة لن يلتفت إليها ، فإذا وجدها مـرة أخرى سيحدد الموقع ، ويوقف البحث .

وتكون نتيجة تنفيذ الاستعلام كما يلى :

INSTR(ADDRESS,'-',1,2)
17
17

ولمعرفة موقع العلامة الثالثة استخدم الجملة التالية :

```
select instr (ADDRESS,'-',1,3) from address;
```

وستكون النتيجة كالتالى :

INSTR(ADDRESS,'-',1,3)
20
20

المثال الثانى والعشرون :

إذا أردنا قطع كلمة من خلال الدالة (SUBSTR) ، بحيث تقوم الدالة بالبحث من خلال تحديد موقعها وفقا لمعطيات الحقل ، ثم نقوم بقطعها . فإن ذلك يستدعى معرفة موقع فى الحقل فى كافة السجلات الموجودة بالجدول .

فعلى سبيل المثال إذا أردنا معرفة موقع الكلمة (Asafra) في الحقل (address) – في كافة السجلات – فإن ذلك يمكن فعله ، من خلال تنفيذ الأمر التالي :

```
select substr (ADDRESS,21,6)  from address;
```

وستكون النتيجة كما يلى :

SUBSTR(ADDRESS,21,
Myam
Asafra

ولاشك أنك لاحظت أن حروف كلمة (Myami) عددها خمسة – وليس ستة أحرف – لكننا كتبنا ستة ، لأن كلمة (Asafra) تتكون من هذا العدد من الأحرف .

أما إذا أردنا قطع كلمة (Asafra) فحسب ، فإن ذلك ممكن من خلال تنفيذ جملة الاستعلام التالية :

```
select substr (ADDRESS,instr (address,'Asafra') ,6)  from
address;
```

أي أننا استبدلنا رقم بداية القطع (21) بالعملية instr (address,'Asafra') ، التي ستحدد لنا مكان الكلمة في السجلين ، أي إن الدالة ستعود برقم يكون هو بداية المقطع .

وبما أننا لن نجد الكلمة في السجل الأول – كما هو معلوم – فسوف تكون القيمة الراجعة صفرا ، بينما توجد قيمة لدى السجل الثاني .

وعلى هذا الأساس فسوف تكون نتيجة تنفيذ الاستعلام كالتالى :

SUBSTR(ADDRESS,INS
Egypt-
Asafra

ومن ناحية أخرى فمن الممكن عدم تحديد عدد الأحرف التي سنقطعها بالأرقام ، إذ يمكننا وضع دالة (Length) بدلاً منها .

أي أن جملة الاستعلام ستكون على النحو التالى :

```
select substr (ADDRESS,instr (address,'Asafra') ,length
('Asafra'))  from address;
```

لتصبح نتيجة تنفيذ الاستعلام هكذا :

```
select substr ( ADDRESS,instr (address,'Asafra') ,length ('Asafra') )
from address;

Execute  Load Script  Save Script  Cancel

SUBSTR(ADDRESS,INS
Egypt-
Asafra

2 rows selected.
```

المثال الثالث والعشرون :

إذا أردت حذف أحد الجداول ، فعليك استخدام الأمر التالي :

```
Drop Table address ;
```

لتجد النتيجة كما يلي :

```
drop table address;

Execute  Load Script  Save Script  Cancel
Table dropped.
```

لاحظ : أن البيانات المسجلة في الجدول تم فقدها عند القيـــام بحـــذف الجدول ، لذا كن على حذر حين تقوم بهذه الخطوة .

المثال الرابع والعشرون :

قد ترغب أحيانا فى لصق كلمتين فـــى حقلـــين مختلفـــين إحـــداهما بألأخرى ، وهذا يعنى الإضافة ، وليس الجمع ، فهو أمر آخر .

فعلى سبيل المثال : إذا كان هناك حقل يحتوي علـــي كلمـــة (sam) والحقل الثاني يحتوي علي كلمة (eh) ، ثم أضفنا كل جزء إلى الآخر ، فسوف تصبح الكلمة (sameh) .

والآن .. نريد عرض موظفي الشركة ، بحيث يقترن الاسم بالوظيفة فى كلمة واحدة ، وفى عمود واحد . وهو مايمكننا عمله مـــن خـــلال تنفيذ جملة الاستعلام التالية :

```
select concat (ENAME,JOB) from emp;
```

وسوف تجد نتيجة تنفيذ الاستعلام كما يلي :

CONCAT(ENAME,JOB)
SMITHCLERK
ALLENSALESMAN
WARDSALESMAN
JONESMANAGER
MARTINSALESMAN
BLAKEMANAGER
CLARKMANAGER
SCOTTANALYST
KINGPRESIDENT
TURNERSALESMAN
ADAMSCLERK
JAMESCLERK
FORDANALYST
MILLERCLERK

14 rows selected.

ففى الشكل السابق يتضح أنه قد تم لصق الكلمتين دون فاصل ، ولكن ذلك قد يسبب حالة كم عدم الفهم ، لذا ينبغى الفصل بين الكلمتين من خلال وضع عبارة (Work AS) بينهما .

```
select concat (concat(ENAME,' Work As '),JOB) from emp;
```

ويجب ملاحظة أننا تركنا مسافة قبل كلمة (Work) ، ومسافة أخرى بعد كلمة (As) . كما استخدمنا الدالتين (Concat) .

وسوف تجد النتيجة كالتالى :

CONCAT(CONCAT(ENAME,'WORKAS'),JOB)
SMITH Work As CLERK
ALLEN Work As SALESMAN
WARD Work As SALESMAN
JONES Work As MANAGER
MARTIN Work As SALESMAN
BLAKE Work As MANAGER
CLARK Work As MANAGER
SCOTT Work As ANALYST
KING Work As PRESIDENT
TURNER Work As SALESMAN
ADAMS Work As CLERK
JAMES Work As CLERK
FORD Work As ANALYST
MILLER Work As CLERK

14 rows selected.

فإذا أردنا تغيير اسم الشكل الشبكي (Alias) بكلمة أخرى فإن ذلـــك ممكن إذا كتبنا جملة الاستعلام التالية :

```
select concat (concat(ENAME,' Work As '),JOB) "Our
Employees" from emp;
```

وتكون النتيجة كالتالى :

Our Employees
SMITH Work As CLERK
ALLEN Work As SALESMAN
WARD Work As SALESMAN
JONES Work As MANAGER
MARTIN Work As SALESMAN
BLAKE Work As MANAGER
CLARK Work As MANAGER
SCOTT Work As ANALYST
KING Work As PRESIDENT
TURNER Work As SALESMAN
ADAMS Work As CLERK
JAMES Work As CLERK
FORD Work As ANALYST
MILLER Work As CLERK

14 rows selected.

المثال الخامس والعشرون :

إذا كنا قد عرفنا أن الدالة (Concat) تستخدم للصق الكلمات ، كما اتضح من المثال السابق ، فيجب أن نعرف أنها لاتساعد فى جمع الحقول مثلما يحدث فى جمع الأرقام .

إذن .. فلنختبر تلك الدالة (Concat) مع الأرقام لإضافة حقل الراتب إلى حقل العلاوة . فذلك سوف يكون من خلال كتابة جملة الاستعلام التالية :

```
select ENAME,concat(SAL,COMM) "Employees's salary"
from emp;
```

وستكون النتيجة كالتالى :

ENAME	Employees's salary
SMITH	800
ALLEN	1600300
WARD	1250500
JONES	2975
MARTIN	12501400
BLAKE	2850
CLARK	2450
SCOTT	3000
KING	5000
TURNER	15000
ADAMS	1100
JAMES	950
FORD	3000
MILLER	1300

14 rows selected.

وهناك طريقة أسهل للصق الكلمات باستخدام العلامة (||) .
ويمكن تنفيذ هذه الطريقة من خلال الاستعلام التالي :

```
select ENAME ||' Work As '|| JOB from emp;
```

حيث النتيجة تكون كالتالى :

ENAME||'WORKAS'||JOB
SMITH Work As CLERK
ALLEN Work As SALESMAN
WARD Work As SALESMAN
JONES Work As MANAGER
MARTIN Work As SALESMAN
BLAKE Work As MANAGER
CLARK Work As MANAGER
SCOTT Work As ANALYST
KING Work As PRESIDENT
TURNER Work As SALESMAN
ADAMS Work As CLERK
JAMES Work As CLERK
FORD Work As ANALYST
MILLER Work As CLERK

14 rows selected.

المثال السادس والعشرون :

إذا أردت إضافة الشهور إلى التاريخ فيمكنك استخدام دالـــة مثـــل : ADD_MONTHS ، حيث تمكنك هذه الدالة من إضافة الـــشهور (بالأرقام) إلي التاريخ المسجل في حقل الجدول ، وهذا يتم – بطبيعة

الحال – أثناء عملية العرض ، لأنها جملة استعلام ، وليست معالجة حقيقية للبيانات ، حتى يمكن حفظها .

فإذا أردت عرض اسم موظف ، مقرونا بتاريخ توظيفه ، ثم أردت عرض عمود آخر يحتوى تاريخ توظيفه ، مع إضافة شهرين لهذا التاريخ ، فإن ذلك ممكن باستخدام الاستعلام التالى :

```
Select
ENAME,HIREDATE,add_months(HIREDATE,2) from emp;
```

وقد قمنا بكتابة الرقم (2) بعد (HireDate) وهو مايعنى إضافة شهرين إلي تاريخ التوظيف .

وتكون نتيجة تنفيذ الاستعلام كما يلى :

ENAME	HIREDATE	ADD_MONT
SMITH	17/12/80	17/02/81
ALLEN	20/02/81	20/04/81
WARD	22/02/81	22/04/81
JONES	02/04/81	02/06/81
MARTIN	28/09/81	28/11/81
BLAKE	01/05/81	01/07/81
CLARK	09/06/81	09/08/81
SCOTT	19/04/87	19/06/87
KING	17/11/81	17/01/82
TURNER	08/09/81	08/11/81
ADAMS	23/05/87	23/07/87
JAMES	03/12/81	03/02/82
FORD	03/12/81	03/02/82
MILLER	23/01/82	23/03/82

14 rows selected.

المثال السابع والعشرون :

من السهل إضافة الأيام إلي التاريخ ، وذلك باستخدام عملية الجمـــع ، حيث تمكننا هذه العملية من إضافة الأيام (بالأرقـــام) إلـــي التـــاريخ المسجل في حقل الجدول . وهذا يتم – أيضا – أثناء عملية العرض ، فهي جملة استعلام (مثلما فى المثال السابق) وليست معالجة حقيقيـــة للبيانات ، بحيث يتم حفظها .

فإذا أردنا عرض سم موظف ، وتاريخ توظيفه ، ثم عرضنا عـــامودا آخر ، يحتوي علي تاريخ التوظيف بعد إضافة شهرين إليه . فإن ذلك يمكن من خلال كتابة الاستعلام التالي :

```
select ENAME,HIREDATE,HIREDATE+2 from emp;
```

لتكون النتيجة كما يلى :

ENAME	HIREDATE	HIREDATE
SMITH	17/12/80	19/12/80
ALLEN	20/02/81	22/02/81
WARD	22/02/81	24/02/81
JONES	02/04/81	04/04/81
MARTIN	28/09/81	30/09/81
BLAKE	01/05/81	03/05/81
CLARK	09/06/81	11/06/81
SCOTT	19/04/87	21/04/87
KING	17/11/81	19/11/81
TURNER	08/09/81	10/09/81
ADAMS	23/05/87	25/05/87
JAMES	03/12/81	05/12/81
FORD	03/12/81	05/12/81
MILLER	23/01/82	25/01/82

14 rows selected.

ويمكنك – أيضا – استخدام الاستعلام السابق لإضافة الشهور ، وذلك بكتابة (30) لإضافة شهر ، أو (60) لإضافة شهرين .. وهكذا .

كما يمكنك – كذلك – طرح الأيام والـشهور ، وذلـك مـن خـلال الاستعلام التالي ، الذى قمنا بطرح يومين فيه :

```
select ENAME,HIREDATE,HIREDATE-2 from emp;
```

فتكون النتيجة كالتالى :

ENAME	HIREDATE	HIREDATE
SMITH	17/12/80	15/12/80
ALLEN	20/02/81	18/02/81
WARD	22/02/81	20/02/81
JONES	02/04/81	31/03/81
MARTIN	28/09/81	26/09/81
BLAKE	01/05/81	29/04/81
CLARK	09/06/81	07/06/81
SCOTT	19/04/87	17/04/87
KING	17/11/81	15/11/81
TURNER	08/09/81	06/09/81
ADAMS	23/05/87	21/05/87
JAMES	03/12/81	01/12/81
FORD	03/12/81	01/12/81
MILLER	23/01/82	21/01/82

14 rows selected.

المثال الثامن والعشرون :

قد تحتاج فى بعض الأحيان إلى استبدال كلمة بأخرى ، كأن تستبدل اسم الموظف برقمه فى الحقل (MGR) ، أو تستبدال اسم القسم الذى يعمل به برقم هذا القسم .

والنموذج التالى يوضح ذلك أكثر :

فى حقل المرتب سوف نستبدل بالرقم (800 (جنيه)) عبارة (Eight Hundred) ، بالرقم (3000) العبارة (Three Thousand) وبالرقم (5000) العبارة (Five Thousand) .

وهذا يتم بطريقتين ، نذكر واحدة منهما فى هذا المثال . حيث يتم ذلك بكتابة الاستعلام التالى :

```
select ENAME,SAL,decode(SAL,800,'Eight
Hundred',3000,'Three Thousand',5000,'Five Thousand')from
emp;
```

ففى الاستعلام السابق حددنا – أولا – الجدول الذى سوف نعمل عليه (وهو SAL) ، ثم حددنا الرقم الذى سنستبدله ، والآخر الذى سنستبدله به ، وفصلنا بينهما بعلامة فاصلة (,) . وهكذا ، عندئذ ، سوف تقوم الدالة (Decode) بالوظيفة المطلوبة .

لتكون نتيجة تنفيذ الاستعلام السابق كما يلى :

ENAME	SAL	DECODE(SAL,800,'EIGHTHUNDRED',3000,'THREET
SMITH	800	Eight Hundred
ALLEN	1600	
WARD	1250	
JONES	2975	
MARTIN	1250	
BLAKE	2850	
CLARK	2450	
SCOTT	3000	Three Thousand
KING	5000	Five Thousand
TURNER	1500	
ADAMS	1100	
JAMES	950	
FORD	3000	Three Thousand
MILLER	1300	

ونريد – الآن – أن نكتب كلمة (Other) في حقول المرتبات الأخرى – خلاف ماحددناه منها

وذلك يتم من خلال تنفيذ الاستعلام التالي :

```
select ENAME,SAL,decode(SAL,800,'Eight
Hundred',3000,'Three Thousand',5000,'Five
Thousand','Other')from emp;
```

وهذه هى النتيجة :

ENAME	SAL	DECODE(SAL,800,'EIGHTHUNDRED',3000,'THREET
SMITH	800	Eight Hundred
ALLEN	1600	Other
WARD	1250	Other
JONES	2975	Other
MARTIN	1250	Other
BLAKE	2850	Other
CLARK	2450	Other
SCOTT	3000	Three Thousand
KING	5000	Five Thousand
TURNER	1500	Other
ADAMS	1100	Other
JAMES	950	Other
FORD	3000	Three Thousand
MILLER	1300	Other

المثال التاسع والعشرون :

فى هذا المثال سوف نتناول الطريقة الثانية ، التى تستطيع من خلالها استبدال كلمة بكلمة أخرى ، أو استبدال رقم بكلمة ، وهو ما أشـــرنا إليه فى المثال السابق.

فإذا أردنا أن نستبدل بــرقم الموظـــف فـــى الحقـــل (MGR) اســم الموظف، أو أردنا أن نستبدل برقم القسم الذى يعمل فيه الموظف بهذا القسم ، فإن ذلك يوضحه النموذج التالى :

في حقل المرتب ، سوف نستبدل بالرقم ((800) جنيه) عبارة (Eight Hundred) ، ونـــستبدل بـــالرقم (3000) (Three Thousand) ، وبالرقم (5000) العبارة (Five Thousand).

فهذا يتم من خلال كتابة الاستعلام التالي ، في مكان العمل :

```
select ENAME,SAL,case SAL when 800 then 'Eight Hundred'
when 3000 then 'Three Thousand' when 5000 then 'Five
Thousand' else 'Other' end from emp;
```

وستكون النتيجة كاالتالى :

ENAME	SAL	CASESALWHEN800THENEIGHTHUNDREDWHEN3000TH
SMITH	800	Eight Hundred
ALLEN	1600	Other
WARD	1250	Other
JONES	2975	Other
MARTIN	1250	Other
BLAKE	2850	Other
CLARK	2450	Other
SCOTT	3000	Three Thousand
KING	5000	Five Thousand
TURNER	1500	Other
ADAMS	1100	Other
JAMES	950	Other
FORD	3000	Three Thousand
MILLER	1300	Other

المثال الثلاثون :

من جديد سننحى (جمل الاستعلام) السابقة جانبا ، ونبدأ ى استعراض مجموعة من الأوامر الجديدة .

فمن أجل حساب متوسط المرتبات يمكننا تنفيذ ذلك من خلال كتابـــة جملة الاستعلام التالية :

```
select avg (SAL) from emp;
```

والنتيجة كالتالى :

AVG(SAL)
2073.21429

.. ومن أجل حساب مجموع المرتبات ، يمكننا تنفيذ ذلك من خــلال جملة الاستعلام التالية :

```
select sum (SAL) from emp;
```

والنتيجة كالتالى :

SUM(SAL)
29025

.. ومن أجل عرض الأكبر من بين المرتبات الموجودة ، يمكننا تنفيذ ذلك من خلال جملة الاستعلام التالية :

```
select max (SAL) from emp;
```

والنتيجة كالتالى :

MAX(SAL)
5000

.. ومن أجل حساب عدد المرتبات الموجودة ، يمكننا تنفيذ ذلك من خلال جملة الاستعلام التالية :

```
select count (SAL) from emp;
```

والنتيجة كالتالى :

COUNT(SAL)
14

.. ومن أجل حساب أصغر المرتبات الموجودة ، يمكننا تنفيذ ذلك من خلال جملة الاستعلام التالية :

```
select min (SAL) from emp;
```

والنتيجة كالتالى :

MIN(SAL)
800

.. ومن أجل حساب عدد السجلات الموجودة فـــي الجـــدول (emp) يمكننا تنفيذ ذلك ، من خلال جملة الاستعلام التالية :

```
select count (*) from emp;
```

والنتيجة كالتالى :

COUNT(*)
14

.. ومن أجل حساب متوسط المرتبات بطريقة جديدة ، يمكننـــا تنفيـــذ ذلك من خلال جملة الاستعلام التالية ، بعـــد تـــسمية العمـــود باســـم (AVG SAL) :

```
select sum(SAL)/count(*) "AVG SAL" from emp;
```

والنتيجة كالتالى :

AVG SAL
2073.21429

المثال الحادى والثلاثون :

ربما تحتاج – أحيانا – إلى عمل حصر بالموظفين الذين يتقاضـــون رواتب أعلى من رقم معين .

فلو أردت – على سبيل المثال – حصر الموظفين الـــذين يتقاضـــى الواحد منهم راتبا يزيد على الف جنيه ، فذلك ممكن بكتابـــة جملـــة الاستعلام التالية . وهى الطريقة العادية .

```
select EMPNO,ENAME from emp where sal >=1000;
```

ولكننا سوف نستخدم طريقة جديـــدة ، كـــم خـــلال (Group By) ، والشرط (Having) .

ومن الممكن أن يتم ذلك من خلال جملة الاستعلام التالية :

```
select EMPNO,ENAME from emp group by
EMPNO,ENAME having  max(sal) >= 1000;
```

لتكون النتيجة كما يلى :

EMPNO	ENAME
7521	WARD
7566	JONES
7844	TURNER
7876	ADAMS
7499	ALLEN
7782	CLARK
7839	KING
7698	BLAKE
7902	FORD
7788	SCOTT
7654	MARTIN
7934	MILLER

12 rows selected

المثال الثانى والثلاثون :

قد يلزمك أحيانا أن تقوم بحصر الموظفين الذين يشتركون فى تقاضى راتب بعينه ، بحيث يتم حصر هؤلاء فى الشركة ، مع إظهار كل مرتب عبى حدة .

وهذا يمكن القيام به ، من خلال استخدام جملة الاستعلام التالية :

```
select count(*),SAL from emp group by SAL;
```

لتكون نتيجة الاستعلام كالتالى :

COUNT(*)	SAL
1	2450
1	5000
1	1300
2	1250
1	2850
1	2975
1	1100
2	3000
1	800
1	1600
1	1500
1	950

فكما تري ، فإن المرتب البالغ 1250 جنيها يتقاضاه اثنان من الموظفين والمرتب البالغ ثلاثة آلاف جنيه يتقاضاه اثنان كذلك . بينما تختلف المرتبات الأخرى ، التى يتقاضاه باقى الموظفين .

المثال الثالث والثلاثين :

قد تريد معرفة الوظائف بالشركة ، أو المرتبات التي يتم صرفها ، أو الأقسام الموجودة بالشركة . وهو مايمكن تنفيذه من خلال الدالة (Distinct) .

فإذا أردت عرض كافة الأقسام الموجودة بالشركة ، دون حدوث تكرار ، فذلك ممكن من خلال جملة الاستعلام التالية :

```
select distinct DEPTNO from emp ;
```

كما يمكن تنفيذ الوظيفة ذاتها ، من خلال كتابة جملة الاستعلام التالية في مكان العمل في أداة iSQL*PLUS :

```
select DEPTNO from emp group by DEPTNO ;
```

لتكون نتيجة تنفيذ الجملتين السابقين كما يلى :

DEPTNO
30
20
10

المثال الرابع الثلاثون :

إذا أردت حصر الموظفين العاملين فى مختلف الأقسام ، بحيث يظهر كل قسم بما يحتويه من موظفيه ، مع مجموع مرتبات موظفى القسم ، وإظهار أكبر الرواتب ، وكذلك متوسط هذه الرواتب فى كل قسم ..

فإن ذلك يمكن تنفيذه من خلال جملة الاستعلام التالية :

```
select DEPTNO,sum(SAl),count(*),max(SAL),avg(SAL) from
emp group by DEPTNO ;
```

فتكون نتيجة تنفيذ الاستعلام كما يلى :

DEPTNO	SUM(SAL)	COUNT(*)	MAX(SAL)	AVG(SAL)
30	[illegible]	[illegible]	2850	1566.66667
20	10875	[illegible]	3000	[illegible]
10	[illegible]	[illegible]	5000	2916.66667

هذا .. ويمكنك إضافة "Employees" بعد count(*) فتكون اسماً للعمود ، كما هو واضح من الجملة التالية :

```
select
DEPTNO,sum(SAl)"",count(*)"Employees",max(SAL),avg(SAL)
from emp group by DEPTNO ;
```

عندئذ ستكون نتيجة تنفيذ الاستعلام كالتالى :

DEPTNO	SUM(SAL)	Employees	MAX(SAL)	AVG(SAL)
30	9400	6	2850	1566.6667
20	10875	5	3000	2175
10	8750	3	5000	2916.6667

3 rows selected.

المثال الخامس والثلاثون :

ربما تود عمل حصر للموظفين كافة فى جميع الأقسام ، وتعرف معظم الوظائف فى كل قسم ، بحيث يظهر كل قسم وبه مايحتويه من وظائف ، ورواتب ، مع إظهار أكبر الرواتب لوظيفة معينة ، ومتوسط الرواتب بكب وظيفة فى كل قسم.

وذلك يمكن تنفيذه من خلال جملة الاستعلام التالية :

```
select count(*),DEPTNO,JOB,SUM(SAL) from emp group by
DEPTNO,JOB ;
```

وتكون النتيجة كما يلى :

COUNT(*)	DEPTNO	JOB	SUM(SAL)
2	20	CLERK	1900
4	30	SALESMAN	5600
1	20	MANAGER	2975
1	30	CLERK	950
1	10	PRESIDENT	5000
1	30	MANAGER	2850
1	10	CLERK	1300
1	10	MANAGER	2450
2	20	ANALYST	6000

ورغم أنه تم تنفيذ الاستعلام إلا أن عدد موظفى كل قسم لم يظهر .. إذن .. فلنستخدم الدالة (Roll Up) ، من خلال تنفيذ جملة الاستعلام التالية :

```
select count(*),DEPTNO,JOB,SUM(SAL) from emp group by
rollup(DEPTNO,JOB)
```

فتكون النتيجة كما يلى :

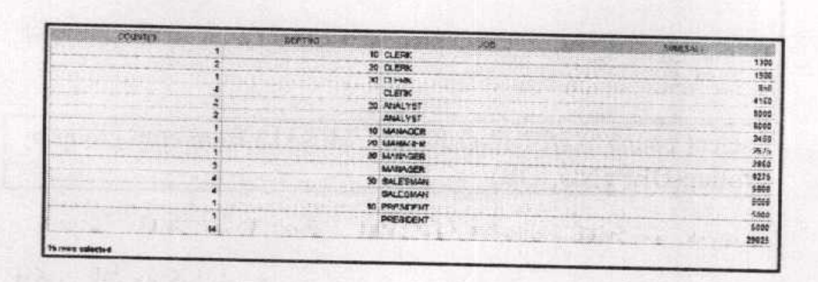
COUNT(*)	DEPTNO	JOB	SUM(SAL)
1	10	CLERK	1300
1	10	MANAGER	2450
1	10	PRESIDENT	5000
3	10		8750
2	20	CLERK	1900
2	20	ANALYST	6000
1	20	MANAGER	2975
5	20		10875
1	30	CLERK	950
1	30	MANAGER	2850
4	30	SALESMAN	5600
6	30		9400
14			29025

13 rows selected

فإذا أردت تنفيذ الجملة السابقة ، مع الوضع فى الإعتبار الترتيب من خلال (JOB) ، فيكفي أن تبدأ بالوظيفة ، كما هو واضح من جملـــة الاستعلام التالية :

```
select count(*),DEPTNO,JOB,SUM(SAL) from emp group by
rollup(JOB,DEPTNO) ;
```

وستكون نتيجة تنفيذ هذا الاستعلام كالتالى :



COUNT(*)	DEPTNO	JOB	SUM(SAL)
1	10	CLERK	1300
2	20	CLERK	1900
1	30	CLERK	950
4		CLERK	4150
2	20	ANALYST	6000
2		ANALYST	6000
1	10	MANAGER	2450
1	20	MANAGER	2975
1	30	MANAGER	2850
3		MANAGER	8275
4	30	SALESMAN	5600
4		SALESMAN	5600
1	10	PRESIDENT	5000
1		PRESIDENT	5000
14			29025

15 rows selected

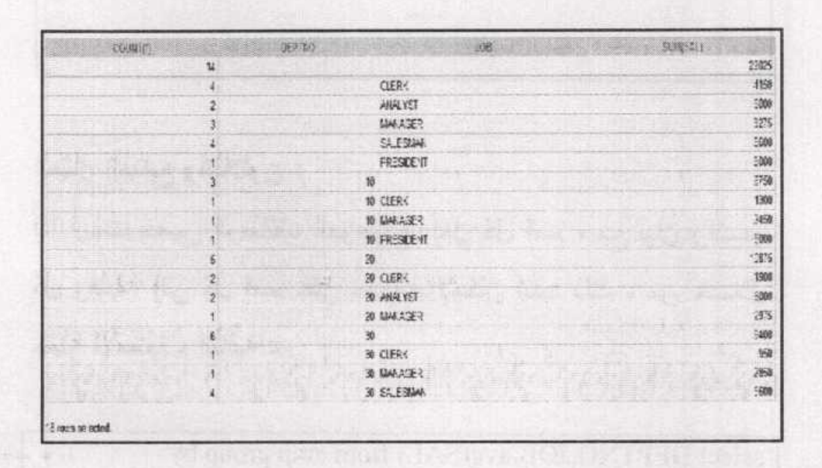
المثال السادس والثلاثون :

فى هذا المثال نريد حصر كافة الموظفين فـــي الأقـــسام ومجمـــوع رواتبهم .. وأكبر المرتبات ... الخ – كما فى المثال السابق ، ولكـــن من خلال استخدام (Cube) ، بحيث يظهر لنا عدد المـــوظفين كافـــة داخل الشركة ، ثم عدد موظفي كل قسم ، مع مجموع رواتبهم .. إلخ

وذلك ممكن باستخدام جملة الاستعلام التالية :

```
select count(*),DEPTNO,JOB,SUM(SAL) from emp group by
cube(DEPTNO,JOB) ;
```

حيث ستكون النتيجة كالتالى :



COUNT(*)	DEPTNO	JOB	SUM(SAL)
14			29025
4		CLERK	4150
2		ANALYST	6000
3		MANAGER	8275
4		SALESMAN	5600
1		PRESIDENT	5000
3	10		8750
1	10	CLERK	1300
1	10	MANAGER	2450
1	10	PRESIDENT	5000
5	20		10875
2	20	CLERK	1900
2	20	ANALYST	6000
1	20	MANAGER	2975
6	30		9400
1	30	CLERK	950
1	30	MANAGER	2850
4	30	SALESMAN	5600

18 rows selected.

ولتنفيذ الجملة السابقة ذاتها ، ولكن ببدء الترتيب من خلال (JOB) ، يكفي البدء بالوظيفة أولاً ، كما تري في جملة الاستعلام التالية :

```
select count(*),DEPTNO,JOB,SUM(SAL) from emp group by
cube(JOB,DEPTNO) ;
```

وستكون النتيجة كالتالى :

COUNT(*)	DEPTNO	JOB	SUM(SAL)
14			29025
4		CLERK	4150
2		ANALYST	6000
3		MANAGER	8275
4		SALESMAN	5600
1		PRESIDENT	5000
3	10		8750
1	10	CLERK	1300
1	10	MANAGER	2450
1	10	PRESIDENT	5000
5	20		10875
2	20	CLERK	1900
2	20	ANALYST	6000
1	20	MANAGER	2975
6	30		9400
1	30	CLERK	950
1	30	MANAGER	2850
4	30	SALESMAN	5600

18 rows selected.

المثال السابع والثلاثون :

إذا أردت حصر الوظائف الموجودة داخل كل قسم ، مع بيان راتـــب كل وظيفة (في كل قسم علي حدة) فبالإمكان تنفيذ ذلك ، من خـــلال جملة الاستعلام التالية :

```
select DEPTNO,JOB,avg(SAL) from emp group by
DEPTNO,JOB,MGR ;
```

وستكون النتيجة كالتالى :

DEPTNO	JOB	AVG(SAL)
30	MANAGER	2850
20	ANALYST	3000
20	CLERK	1100
30	CLERK	950
10	MANAGER	2450
20	CLERK	800
20	MANAGER	2975
10	PRESIDENT	5000
10	CLERK	1300
30	SALESMAN	1400

10 rows selected.

والآن .. سوف نتناول الـ (Grouping Sets) ، التى تختلف عـن الـ (Roll up) ، كما تختلف عن الـ (Cube) ، فى أنها لا تحتوي على مجموع (Totals) .

وذلك من خلال جملة الاستعلام التالية :

```
select DEPTNO,JOB,avg(SAL) from emp group by grouping
sets ((DEPTNO,JOB,MGR),(DEPTNO,MGR),(JOB,MGR)) ;
```

وتكون نتيجة تنفيذ الاستعلام السابق كما يلى :

DEPTNO	JOB	AVG(SAL)
30	CLERK	950
10	CLERK	1300
20	CLERK	1100
20	CLERK	800
20	ANALYST	3000
10	MANAGER	2450
20	MANAGER	2975
30	MANAGER	2850
30	SALESMAN	1400
10	PRESIDENT	5000
	CLERK	950
	CLERK	1300
	CLERK	1100
	CLERK	800
	ANALYST	3000
	MANAGER	2758.33333
	SALESMAN	1400
	PRESIDENT	5000
20		2975
10		2450
30		1310
10		1300
20		3000
20		800
DEPTNO	JOB	AVG(SAL)
10		5000
20		1100
30		2850

27 rows selected.

المثال الثامن والثلاثون :

سوف نتناول فى هذا المثال كيفية العمل مع جدولين مختلفين . وذلك بعرض الموظفين والأقسام التى يعملون بها .

فاسم القسم موجود في الجــدول (dept) ، بينمــا الجــدول (emp) يحتوي علي رقم القسم فحسب ، والرقم هو الذى يشير إلي اسم قسم .

ولتنفيذ ذلك قم بكتابة جملة الاستعلام التالية :

```
select ENAME,DNAME from emp,dept where
emp.deptno=dept.deptno ;
```

وهذه هى نتيجة الاستعلام :

ENAME	DNAME
SMITH	RESEARCH
ALLEN	SALES
WARD	SALES
JONES	RESEARCH
MARTIN	SALES
BLAKE	SALES
CLARK	ACCOUNTING
SCOTT	RESEARCH
KING	ACCOUNTING
TURNER	SALES
ADAMS	RESEARCH
JAMES	SALES
FORD	RESEARCH
MILLER	ACCOUNTING

14 rows selected.

المثال التاسع والثلاثون :

نريد فى هذا المثال عرض كل موظف ، وراتبه الذي يحصل عليه ، رغم أن درجة المرتب موجودة في جدول آخر هو (salgrade) الذى يشار فيه إلى درجة راتب بعينه ، فى حين أن الجدول (emp) يحتوي علي الراتب فحسب .

وذلك يكون من خلال الإستعلام التالى :

```
select ENAME,SAL,GRADE from emp,salgrade where SAL
between LOSAL and HISAL ;
```

وسوف تكون النتيجة كالتالى :

ENAME	SAL	GRADE
SMITH	800	1
JAMES	950	1
ADAMS	1100	1
WARD	1250	2
MARTIN	1250	2
MILLER	1300	2
TURNER	1500	3
ALLEN	1600	3
CLARK	2450	4
BLAKE	2850	4
JONES	2975	4
SCOTT	3000	4
FORD	3000	4
KING	5000	5

14 rows selected

المثال الأربعون :

فى هذا المثال نريد عرض الموظفين كافة ، مع عرض القسم الـــذى يعمل فيه كل موظف ، إضافة إلى الأقسام التى ليس بها موظفين .

ويمكننا تنفيذ ذلك من خلال جملة الاستعلام التالية :

```
select ENAME,DNAME from emp,dept where
emp.deptno(+)=dept.deptno ;
```

وكما ترى فى الاستعلام السابق ، فإن وجود علامة (+) فـــي مكـــان الموظفين (emp.deptno) موجودة ، لأن النقص فـــي المـــوظفين ، بينما الزيادة في الأقسام ، لذلك نضع هـــذه العلامـــة ، فـــى المكـــان المذكور ، كما هو واضح فى الإستعلام السابق :

وإذن .. فستكون النتيجة كما يلى :

ENAME	DNAME
CLARK	ACCOUNTING
KING	ACCOUNTING
MILLER	ACCOUNTING
JONES	RESEARCH
FORD	RESEARCH
ADAMS	RESEARCH
SMITH	RESEARCH
SCOTT	RESEARCH
WARD	SALES
TURNER	SALES
ALLEN	SALES
JAMES	SALES
BLAKE	SALES
MARTIN	SALES
	OPERATIONS

وكما ترى فـــإن ن القـــسم (OPERATIONS) لا يحتـــوي علـــي موظفين .. ولكن .. إذا أردنا إضافة بيانات موظف جديد ليعمل فـــي القسم (OPERATIONS) ، فسوف يتم ذلك من خلال مايلى :

توجـــه – أولا – إلـــى جـــدول الأقـــسام لمعرفـــة رقـــم القـــسم (OPERATIONS) ، وسوف تجد أنه الرقم (40) .

إذن .. قم بتنفيذ الجملة التالية لإدخال بيان جديد إلى الجدول :

```
insert into emp values
(7950,'Osama','MANAGER','7839','15/12/07',3500,null,40);
```

وهذه نتيجة أمر المعالجة :

```
1 row created.
```

ويمكنك التأكد من نجاح عملية الإدخال بتجربـــة جملـــة الاســـتعلام select * from emp ، لكى تستعرض الموظفين ، وسوف تجد أن الموظف الجديد قد ظهر في الشكل الشبكي :

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7950	Osama	MANAGER	7839	15/12/07	3500		40

وعند تنفيذ الاستعلام التالي ستجد أن القسم (OPERATIONS) يحتوي علي الموظف الجديد :

```
select ENAME,DNAME from emp,dept where
emp.deptno(+)=dept.deptno ;
```

أى أن القسم أصبح على النحو التالى :

ENAME	DNAME
KING	ACCOUNTING
MILLER	ACCOUNTING
CLARK	ACCOUNTING
SMITH	RESEARCH
FORD	RESEARCH
ADAMS	RESEARCH
SCOTT	RESEARCH
JONES	RESEARCH
JAMES	SALES
ALLEN	SALES
BLAKE	SALES
MARTIN	SALES
TURNER	SALES
WARD	SALES
Osama	OPERATIONS

15 rows selected.

المثال الحادى والأربعون :

نريد فى هذا المثال عرض كافة الموظفين ، مع عرض كل قسم يعمل فيه أحدهم ، وكذلك عرض الموظفين ، الذين ليس لهم قسم خاص بهم.

وهذا يتم من خلال كتابة جملة الاستعلام التالية :

```
select ENAME,DNAME from emp,dept where
emp.deptno=dept.deptno(+) ;
```

وكما ترى فى الاستعلام السابق ، فإن وجود علامة (+) فـــي مكـــان الأقسام (dept.deptno) ، وذلك لوجود النقص فى الأقسام ، بينمـــا الزيادة في الموظفين ، ذلك نضع هذه العلامة فى المكان المذكور .

وستكون نتيجة هذا الاستعلام كما يلى :

ENAME	DNAME
SMITH	RESEARCH
ALLEN	SALES
WARD	SALES
JONES	RESEARCH
MARTIN	SALES
BLAKE	SALES
CLARK	ACCOUNTING
SCOTT	RESEARCH
KING	ACCOUNTING
TURNER	SALES
ADAMS	RESEARCH
JAMES	SALES
FORD	RESEARCH
MILLER	ACCOUNTING

14 rows selected.

ومن الشكل السابق يمكنك ملاحظة لأن كل الموظفين يعملون داخـــل أقسام ، وليس هناك أى موظف لا يعمل في أحد الأقسام ، لذا فسوف

نقوم بإنشاء موظف لا يعمل في أى قسم ، ثم ننفذ هذا الاستعلام مرة أخرى .

ويمكن عمل ذلك من خلال جملة الاستعلام التالية :

```
insert into emp values
(7960,'Muhammad','MANAGER','7839','05/02/07',3700,null,null);
```

وقد كتبنا (Null) مرتين : الأولي مــن أجــل حقــل (COMM) ، والثانية من أجل حقل (DEPTNO) .

والآن .. قم بتنفيذ جملة الاستعلام التالية ، لتري الموظفين الــذين لا ينتمون إلى أى قسم :

```
select ENAME,DNAME from emp,dept where
emp.deptno=dept.deptno(+) ;
```

وستجد النتيجة كالتالى :

ENAME	DNAME
Osama	OPERATIONS
Muhammad	
SMITH	RESEARCH
ALLEN	SALES
WARD	SALES
JONES	RESEARCH
MARTIN	SALES
BLAKE	SALES
CLARK	ACCOUNTING
SCOTT	RESEARCH
KING	ACCOUNTING
TURNER	SALES
ADAMS	RESEARCH
JAMES	SALES
FORD	RESEARCH
MILLER	ACCOUNTING

16 rows selected.

ولكن .. عند عرض الموظفين كافة ، وعرض كل قسم يعمل فيه موظف لن تجد الموظف (Muhammad) .

ولكى تري ذلك الموظف قم بتنفيذ جملة الاستعلام التالية :

```
select ENAME,DNAME from emp,dept where
emp.deptno=dept.deptno ;
```

وسوف تكون النتيجة كالتالى :

ENAME	DNAME
Osama	OPERATIONS
SMITH	RESEARCH
ALLEN	SALES
WARD	SALES
JONES	RESEARCH
MARTIN	SALES
BLAKE	SALES
CLARK	ACCOUNTING
SCOTT	RESEARCH
KING	ACCOUNTING
TURNER	SALES
ADAMS	RESEARCH
JAMES	SALES
FORD	RESEARCH
MILLER	ACCOUNTING

15 rows selected.

الفصل الثالث

الأوراكل مع الفيجوال بيسيك 6

Oracle With Visual Basic 6

الأوراكل مع الفيجوال بيسيك Oracle with Visual Basic

تحدثنا في الفصلين السابقين عن كيفية التعامل مع قواعد البيانات أوراكل ، وطرق معالجتها ، وغير ذلك ، لكننا لم نتطرق إلي كيفية معالجة تلك البيانات من خلال واجهه برنامج مجمع ، عبر لغة مثل الفيجوال بيسك .

ولأن هذا الكتاب لا يتسع لكل ما يمكن قوله عن الفيجوال بيسيك فسوف نتناول لغة (الفيجوال بيسك 6) فحسب . فإذا أردت المزيد فعليك بإقتناء كتاب (كل شيء عن أوراكل) من اصدار دار البراء ، فهمو يتناول كيفية معالجة قاعدة بيانات أوراكل ، من خلال لغة الفيجوال بيسيك 2005 ، و (سى شارب 2005) ، كما يتناول تطبيقات الويب (ASP.NET 2.0) .

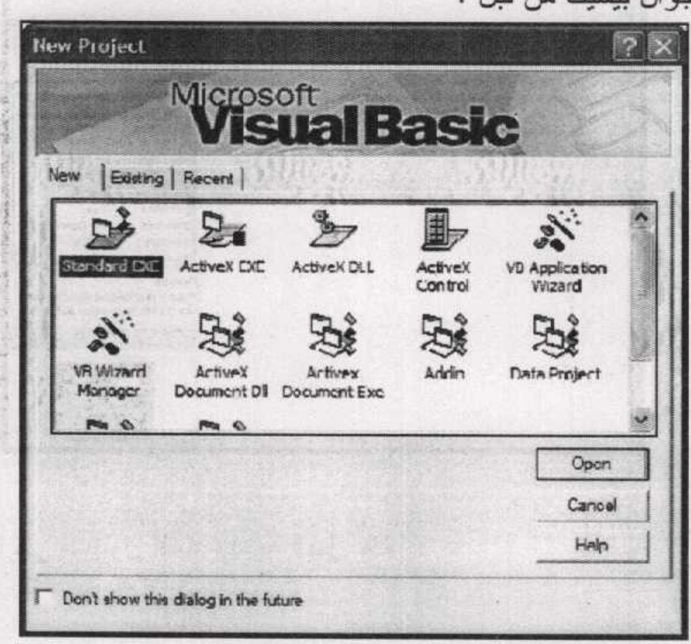
ولنبدأ الآن فى موضوعنا .

تعد لغة (فيجوال بسيك 6) من أشهر لغات البرمجة فى العالم، كما أن كثيرا من المستخدمين بدءوا عملهم من خلال هذه اللغة ، السلسلة . وهى التى مهدت لنجاح (ميكروسوفت) . وأصبحت هناك بيئة جديدة للتطوير ، هى (NET).

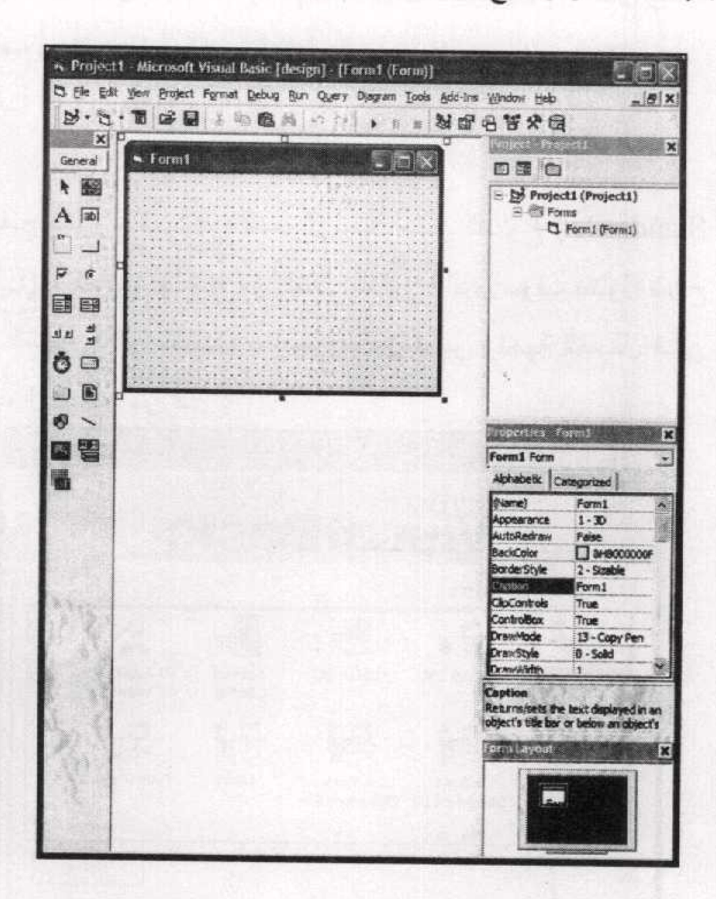
والآن .. قم بتركيب الفيجوال استوديو (أو الفيكوال بيسيك) فحسب ، ومن الأفضل أن تقوم بتركيب (Visual Basic SP6) . الذى يمكنك الحصول عليه من خلال موقع دار البراء ، ضمن ملفات الدعم .

ثم أبدأ العمل معنا ..

قم بفتح مشروع جديد في الفيجوال بيسك مـــن النـــوع (Standard EXE) ، كما هو واضح من الشكل التالى ، الذى سوف نتناول شرح أهم أقسامه بشكل مختصر ، لمساعدة من لم ير واجهة العمـــل فـــى الفيجوال بيسيك من قبل :



وقد ظهرت لنا واجهة العمل الإفتراضية ، حين قمنا بإنشاء المشروع الجديد ، كما هو واضح من الشكل التالى :



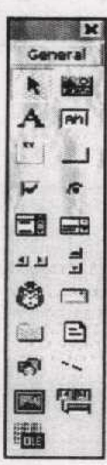
وفيما يلى سنتناول أقسام واجهة العمل هذه ..

صندوق الأدوات :

هذا الصندوق يسمح لنا بإدراج أية أداة نريدها علي النمــوذج الذي نقوم بتصميمه. فهو يحتوى على الأدوات التالية :

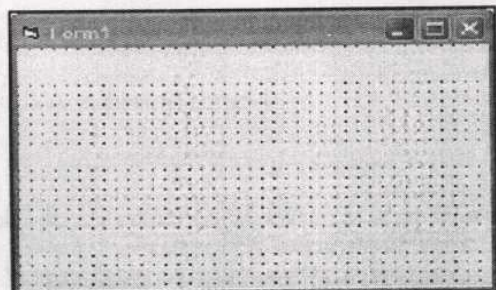
- أداة العنوان : Label
- أداة النص : Text
- أداة الزر : Command
- أداة البيانات : DataControl

كما يحتوى على أدوات بيانات أخرى متقدمــة مثــل (ADODC) ، وأداة عرض البيانات الشبكية (Flex Grid) ، وأدوات أخرى كثيــرة لم نذكرها . فقد ذكرنا من الأدوات ماسوف نحتاج اليه فى مــشروعنا القادم فحسب :



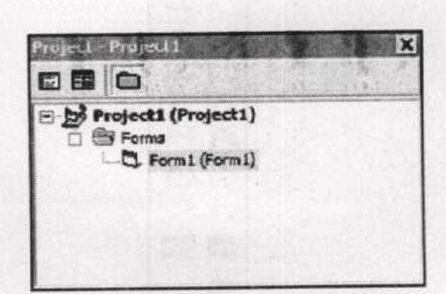
تصميم النموذج :

يمكنك رسم الأدوات وتصميمها علي النموذج التالى ، الـذى يتيح لك تنفيذ الوظائف المطلوبة ، حيـث تـستطيع إدراج أيـة أداة موجودة في صندوق الأدوات ، لتظهر علي شكل نموذج ، كما فـى الشكل السابق الخاص بصندوق الأدوات ، لتوظيفها كما تشاء :



متصفح المشروع :

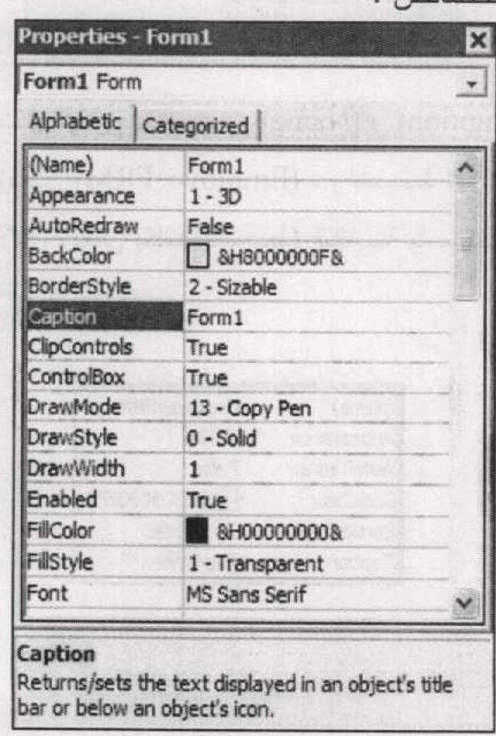
يحتوي متصفح المـشروع علـي النمـاذج ، والوحـدات ، والفصائل ، وسائر الملفات المصاحبة للمشروع ، كما ترى فى الشكل التالى :



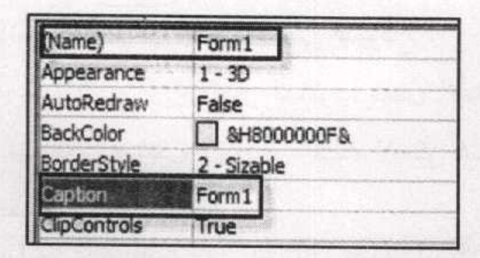
نافذة الخصائص :

لابد من معرفة هذا الجزء بشكل واضح .. إذ أنك حين تقوم بإنشاء أداة نص (Text) تضغط بزر الفأرة الأيمن ضغطا مزدوجـــا (Double Click) ، لتتم إضافة الأداة علي النموذج .

ولكن.. لو أنك تريد تغيير لون النص المكتوب داخل الأداة ، أو لون الأداة نفسها ، ليتوافق مع شكل النموذج ، فذلك لايمكن إتمامه إلا من خلال نافذة الخصائص .



فلو أردت – على سبيل المثال – تغيير اسم النموذج وعنوانه ، فعليك الضغط فى مكان التصميم .. ثم توجه إلي نافذة الخصائص :



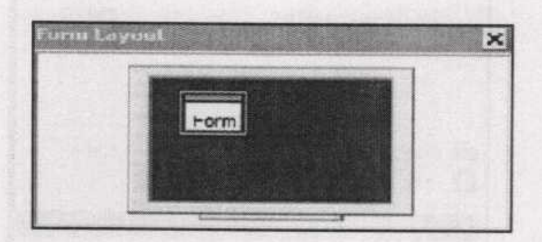
(Name)	Form1
Appearance	1 - 3D
AutoRedraw	False
BackColor	&H8000000F&
BorderStyle	2 - Sizable
Caption	Form1
ClipControls	True

وترى فى الشكل السابق خاصـيتي (Name) و (Caption) ، فقـم بـضبط الأولـي (EmployeeFRM) ، واضـبط الثانيـة علـي (Employee) . ونعنى بكلمة (ضـبط) الكتابـة باسـتخدام لوحـة المفاتيح:

(Name)	EmployeeFRM
Appearance	1 - 3D
AutoRedraw	False
BackColor	&H8000000F&
BorderStyle	2 - Sizable
Caption	Employee

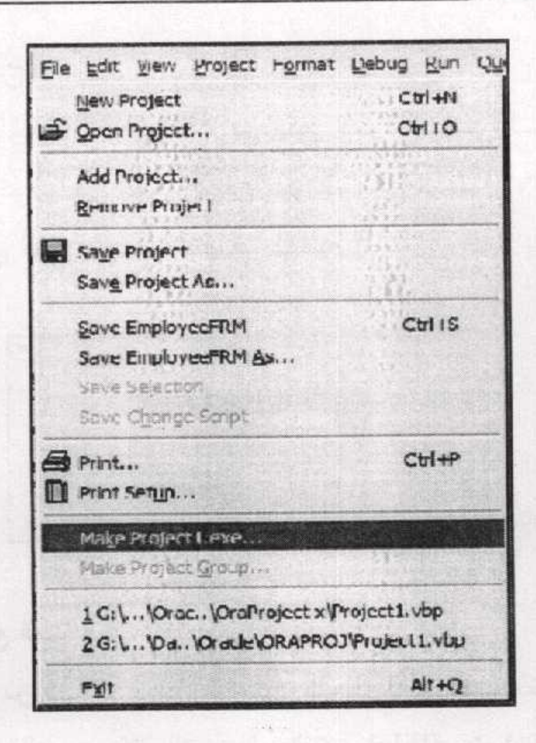
موقع النموذج :

هذه النافذة تسمح بالتحكم في مكان النموذج علـــي الــشاشة ، بحيث يمكن جعلها فى أقصي اليمين ، أو أقــصي اليــسار ، أو فـــي أعلى، أو في أسفل ، أو في منتصف الشاشة :



كيفية تجميع المشروع :

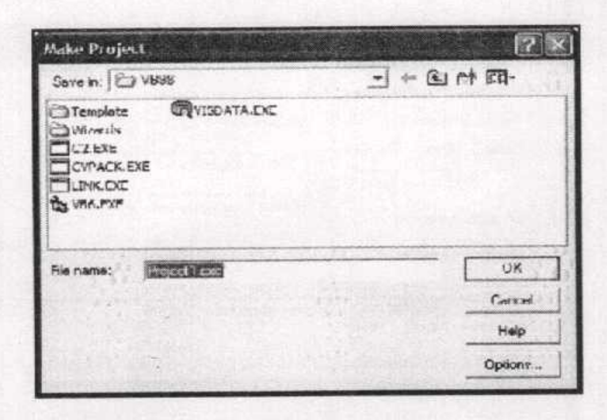
بعد انتهائك من تصميم المشروع يمكنك تحويله إلـــي ملــف تنفيــذي (EXE) ، وذلك من خلال التوجه إلي القائمة (File) ، ثم (Make project1.exe) :



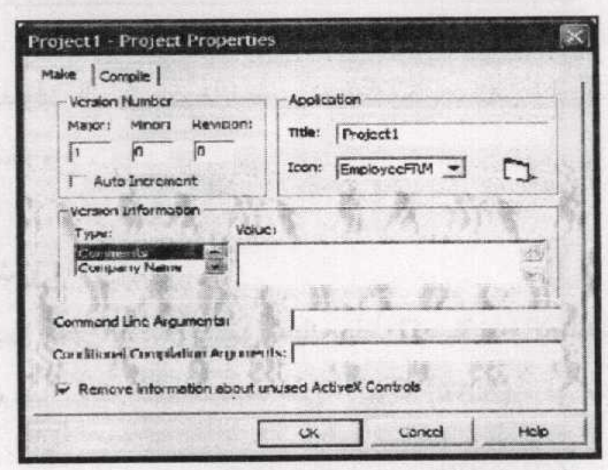
والبرنامج يطلب منك اختيار مجلد التجميع ، فقد أصبح لـك ملفـك التنفيذي .

ولكن لنهتم الآن بأمر آخر ..

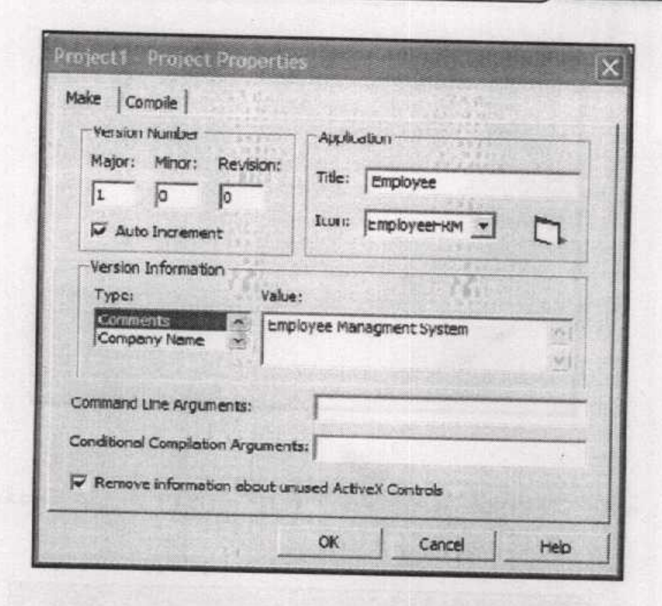
فعند توجهك إلي القائمة (File) ، ثم (Make project1.exe) سوف تظهر لك النافذة التالية :



فإذا ضغطت على الزر(Options) ، سوف تظهر لك النافذة التالية :



فقم بكتابة (Employe) في الخانة (Title) ، واكتب ما تريده ، فيما يتعلق بالتطبيق الذي تعمل عليه ، وذلك في الخانة (Comments) :

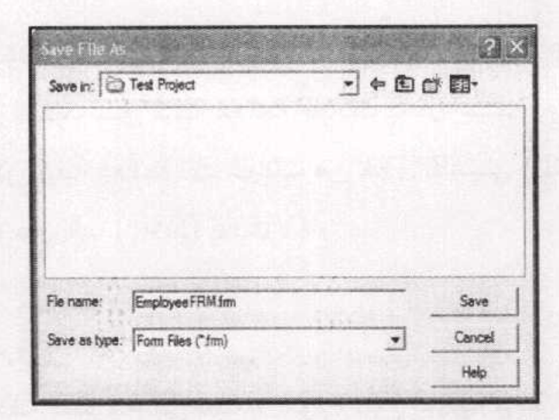


وتلك الخطوات السابق ذكرها تختص بضبط الخـــصائص والبيانـــات العامة للمشروع .

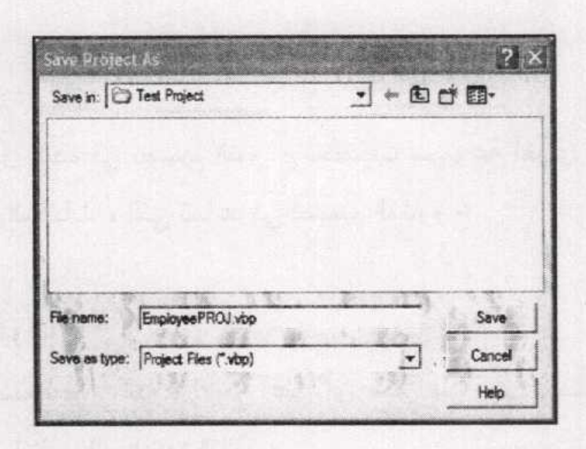
حفظ المشروع :

بين الحين والآخر ينبغى عليك حفظ التعديلات والإضافات ، التي تقوم بها . ويتم ذلك بالضبط علي الأيقونة (Save) ، في شريط الأدوات :

وعند إجراء الحفظ للمرة الأولى سوف تظهر لك النافذة التالية ، لتحدد – من خلالها – مكان حفظ النموذج :



ثم تظهر لك النافذة التالية لتحديد مكان حفظ ملف المشروع نفسه :



مشروع موظفي الشركة Employee Management System

تناولنا فيما سبق كيفية معالجة البيانات بشكل عـــام . ولكننـــا سنتناول الآن كيفية معالجة تلك البيانات من خلال تقنيتـــي (SQL) و (ADO) ، عبر الـــ (Visual Basic) .

هدف المشروع :

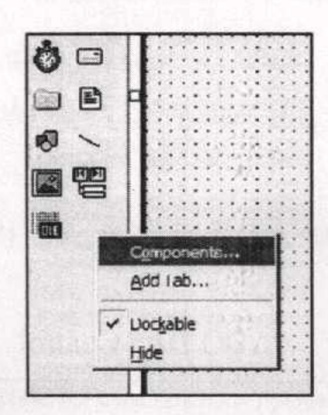
يهدف المشروع إلى تصميم نموذج يقوم معالجة بيانات جدول الموظفين (EMP) ، من خلال التعديل ، والإضـــافة ، والحـــذف ، والبحث .

بدء العمل :

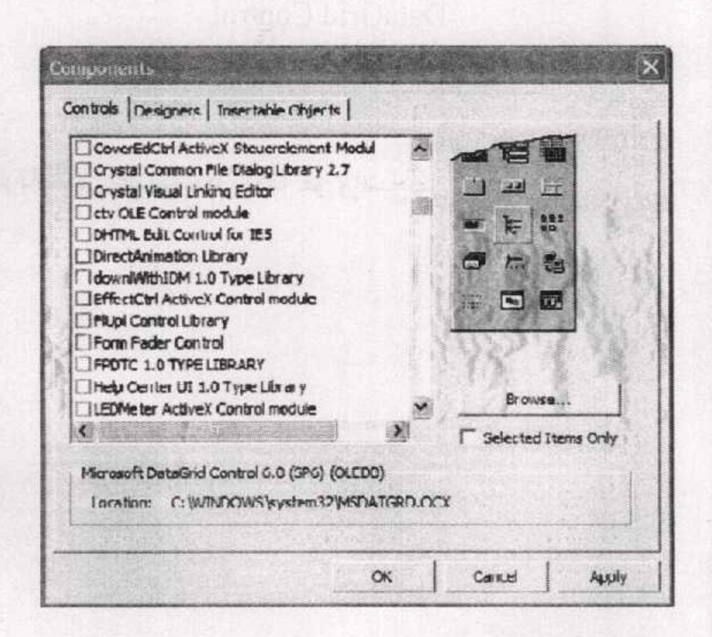
قبل البدء في تصميم النموذج سنضيف مجموعة أخرى مـــن الأدوات والمكتبات ، التي تساعد في تصميم المشروع .

إضافة الأدوات :

اضغط بزر الفارة الأيمن علي أي مكان فارغ في صـــندوق الأدوات ، لتظهر لك النافذة التالية :



فاختر منها (Components) ، وعندئذ ستظهر لك النافذة التالية :



تسمح لك هذه النافذة بإضافة مجموعة من الأدوات ، التي لا تظهــر تلقائياً علي صندوق الأدوات ، وذلك لاستخدامها في التطبيــق الــذي تعمل عليه .

انزل باللفافة (Scroll) ، حتى تصل إلي الأداتين التاليتين :

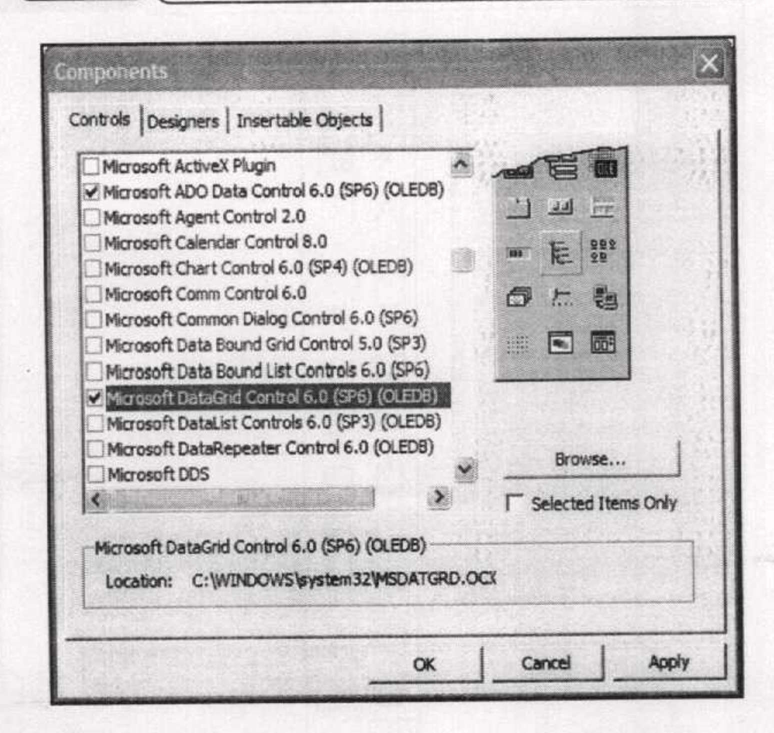
ADO Data Control

Microsoft ADO Data Control 6.0 (SP6) (OLEDB)

DataGrid Control

Microsoft DataGrid Control 6.0 (SP6) (OLEDB)

ثم قم بالتأشير عليهما معا . كما هو واضح من الشكل التالى :



وسوف تجد انه تمت إضافة الأداتين إلي صندوق الأدوات :

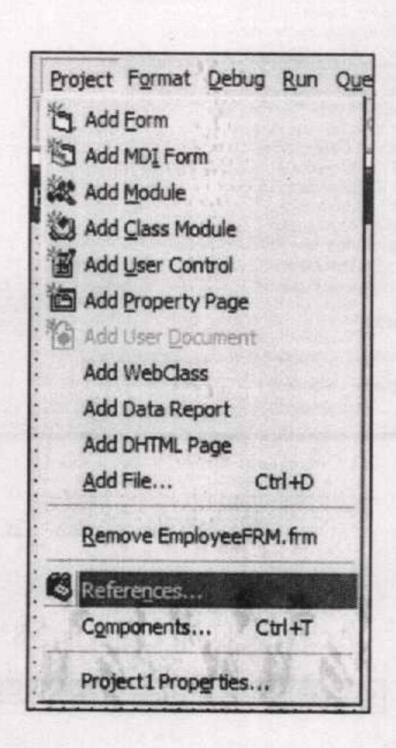


إضافة (References) :

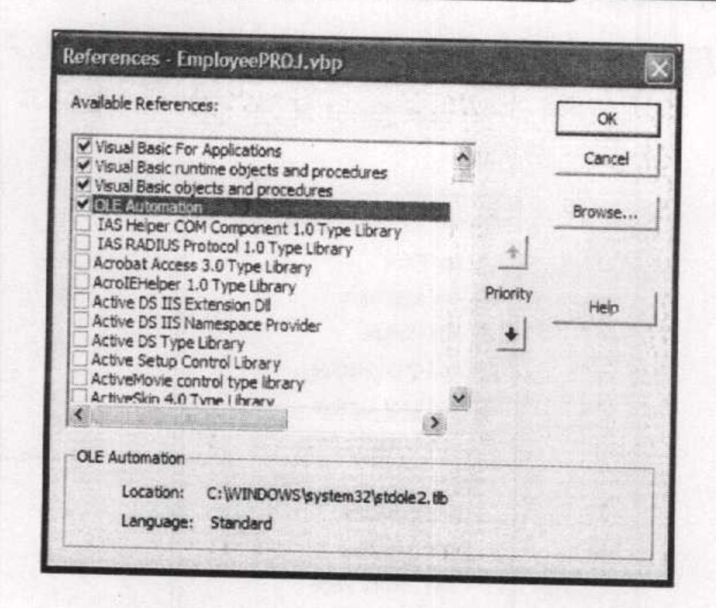
تساعدنا المكتبات علي كتابة كود أقصر ، كما تساعدنا من خلال الأكواد الفرعية غير الظاهرة .

فالمكتبة (Oracle inproc Server) – على سبيل المثال – تساعدنا فى التعامل مع قاعدة البيانات (من النوع أوراكل) بشكل اسرع وأبسط ، دون حاجة لكتابه أكواد كبيرة .

ولكى تتوجه إلى نافذة إضافة المكتبات اضغط علــي (references) في القائمة (Project) ، كما هو واضح من الشكسل التالى :



وسوف تظهر لك النافذة (references) التالية :



انزل إلى أسفل ، حتى تجد المكتبة التالية :

ActiveX Data Objects 2.8

Microsoft ActiveX Data Objects 2.8 Library

Oracle InProc Server 5.0

Oracle InProc Server 5.0 Type Library

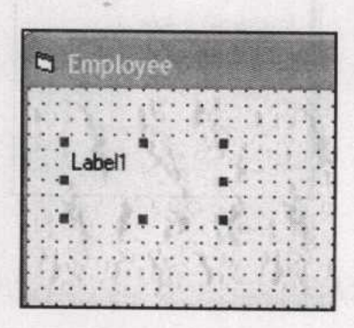
وحيث إن هناك أكثر من إصدارة من المكتبة الأولي ، لذا يجب عليك اختيار المكتبة (2.8) بوجه خاص :

تصميم النموذج :

سوف نبدأ الآن في وضع الأدوات على النمــوذج ، ولنبــدأ بوضع أدوات العناوين..

أداة العنوان : A

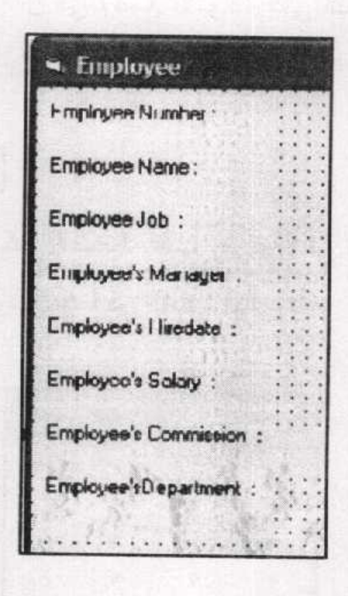
هذه الأداة يكفي الضغط عليها (مرتين) قي صندوق الأدوات؛ لتظهر لك علي النموذج ،كما فى الشكل التالى :



- توجه– الآن– إلي نافــذة الخــصائص،واكتب (Employee Number Caption)

Caption | Label1

ثم قم بإضافة سبع أدوات أخرى إلي النموذج ، وقم بضبط خواصـــهم ليكونوا مناسبين لما فى الشكل التالى :



توجه إلي صندوق الأدوات ، وقم بإدراج ثمانى أدوات نص (Text):

ab|

واضبط خاصية (Name) في أدوات النص ، على النحو التالى :

EMPNOTXT
ENAMETXT
JOBTXT
MGRTXT
HIREDATETXT
SALTXT
COMMTXT
DEPTNOTXT

ثم قم بمسح ما هو مكتوب في خاصية (Text) ، بحيث يصبح محتوي أداة النص فارغا . بعدها ضع أداة النص أمام العنوان المناسب لها ، فتضع أداة النص (EMPNOTXT) أمام أداة عنوان (Employee Number) وهكذا..

وحاول أن تجعل مظهر النموذج قريبا مما فى الشكل التالى :

Employee

Employee Number :
Employee Name :
Employee Job :
Employee's Manager :
Employee's Hiredate :
Employee's Salary :
Employee's Commission :
Employee'sDepartment :

توجه – مرة أخرى – إلي صندوق الأدوات ، ثم أضف أربـــع أدوات زر إلي النموذج :

واضبط خاصية (Name) ، وفقا للترتيب التالي :

SAVEBCMD
NEWCMD
FINDCMD
DELCMD

ثم خاصية (Caption) ، وفقا للترتيب التالي :

Save
NEW
Search
Delete

وقم بوضعها في المواقع التى تراها فى الشكل التالى :

Employee

Employee Number :
Employee Name :
Employee Job
Employee's Manager :
Employee's Hiredate :
Employee's Salary :
Employee's Commission :
Employee's Department

Delete | Search | NEW | Save

إضافة أداة البيانات :

توجه الآن إلي الأداتين ، اللتين سبق أن أضفناهما إلي صندوق الأدوات . واضغط (مرتين) علي الأداة (ADODC) :

والآن .. قم بضبط خاصية (Name) إلي EmployeeADODC ، وخاصية (Caption) إلي (Employee) . وضعهما في الموقع الخاص بهما، كما هو موضح من الشكل التالى :

Employee

Employee Number :

Employee Name :

Employee Job :

Employee's Manager :

Employee's Hiredate :

Employee's Salary :

Employee's Commission :

Employee'sDepartment :

Delete | Search | NEW | Save

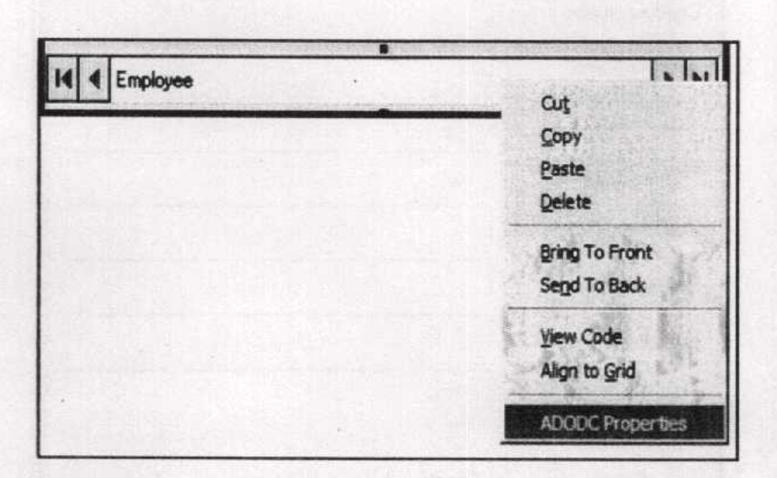
Employee

ويجب أن تلاحظ أن هذه الأداة تسمح لك بالتوجه إلي السجل الـسابق والسجل التالي ، كما تسمح بالتوجه إلى أول السجلات وآخرها .

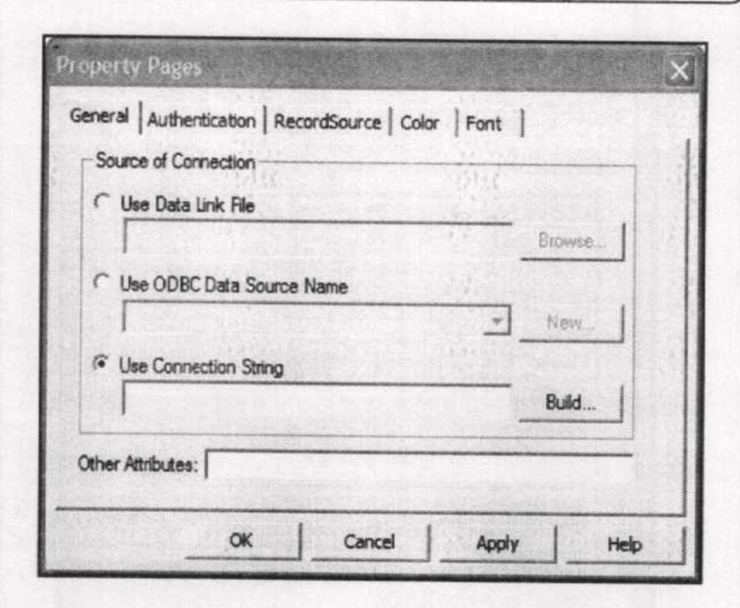
الاتصال مع قاعدة بيانات أوراكل

سنقوم الآن بربط التطبيق الذي نعمل عليه بقاعدة البيانـات ، ومن ثم نربطهما بجدول الموظفين .

- ولكى تقوم بذلك اضغط بـزر الفـأرة الأيمـن علـي الأداة EmployeeADODC علي النموذج :



وسوف تظهر لك النافذة التالية ، التـى تخـص ضـبط خـصائص الاتصال بقواعد البيانات مختلفة الأنواع :

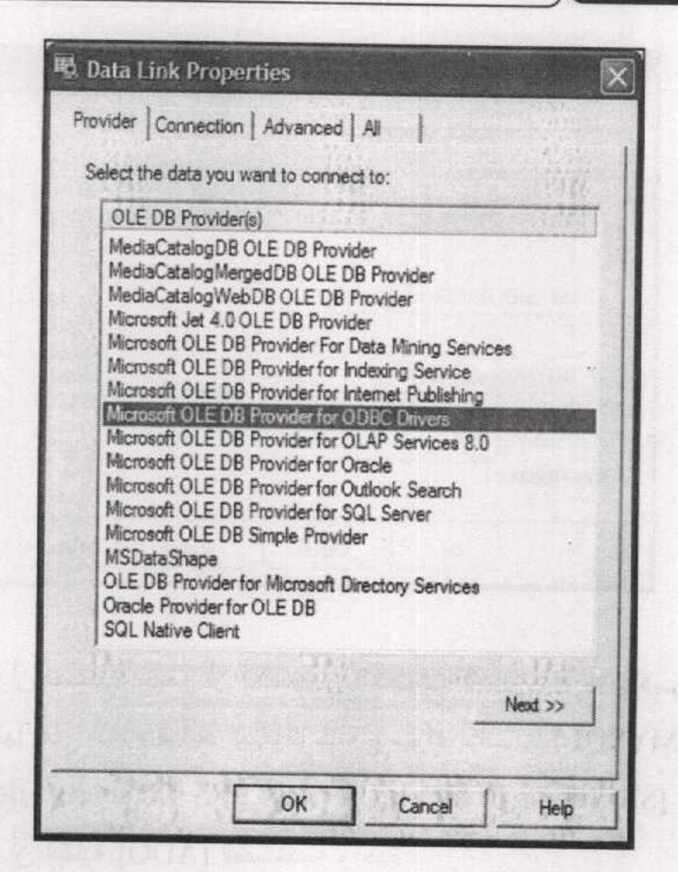


وقبل أن نستطرد ينبغى أن تلاحظ أن الخطوات السابق ذكرها يمكـن تطبيقها على أنواع قواعد البيانات كافة ، سواء أكانـت (MYSQL) أو (ACCESS Database File) أو (SL Server Database) ، حيث أن التقنية (ADO) المستخدمة واحدة .

ولنعد الآن إلي عملنا ..

- اضغط على الزر (Build) ، الذي تراه أمامك فى الـشكل السابق :

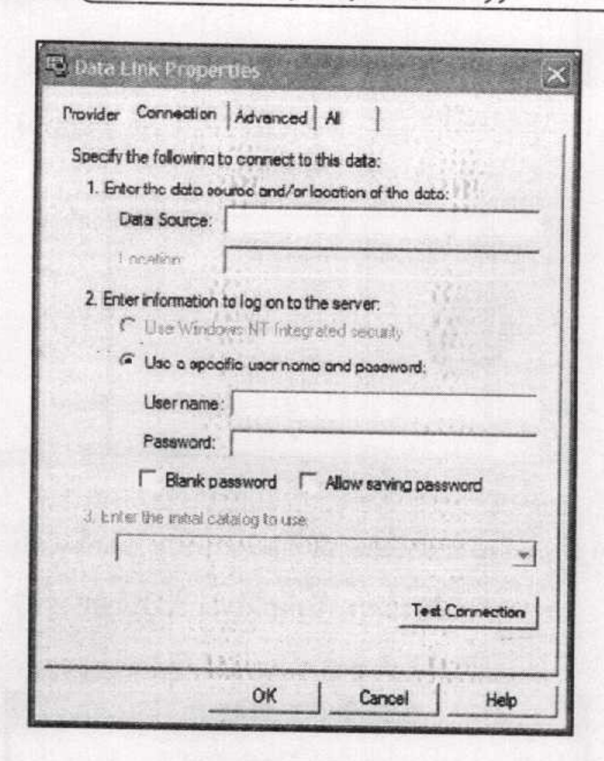
وستظهر لك النافذة التالية ، الخاصة بربط قاعدة البيانات :



فقم باختيار Oracle Provider for OLE DB .. ثم اضغط Next

Oracle Provider for OLE DB

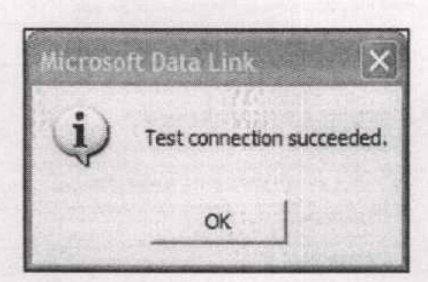
والآن.. سنقوم – معا – بوضع اعتبارات الدخول والاتصال :



- وفي القسم الخاص بكلمة مرور إلى قاعدة البيانـــات اكتـــب (Scott) ، وكلمة المرور الخاصة بـــه ، ولابـــأس باختيـــار (Save Password) :

Use a specific user name and password:
User name: scott
Password: *****

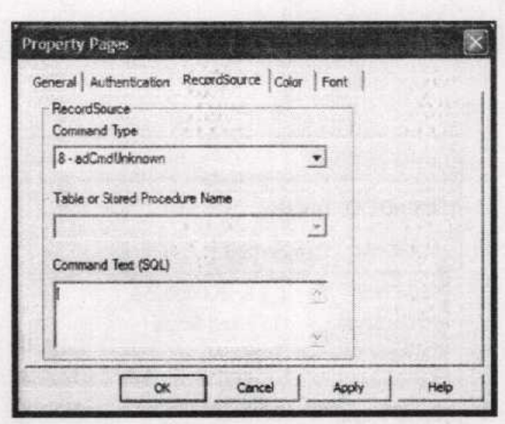
- اختر الآن الاتصال ، مــن خــلال الــضغط علــي الــزر (TestConnection) ، وسوف تظهر لك الرسالة التالية :



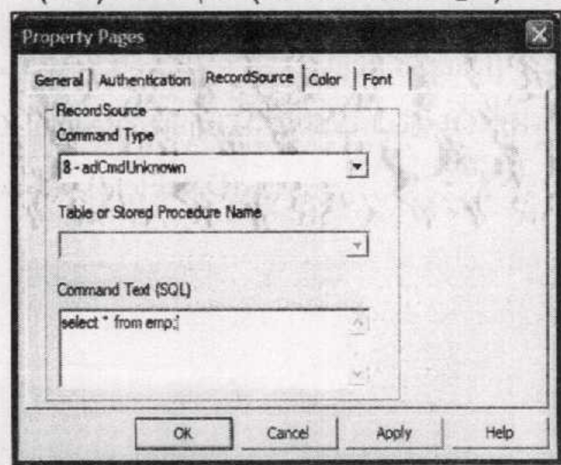
- فاضغط علي (ok) ، لتعود إلى نافذة الخــصائص الرئيــسية لأداة EmployeeADODC :

Property Pages
General | Authentication | RecordSource | Color | Font
Source of Connection
Use Data Link File
Browse...
Use ODBC Data Source Name
New...
Use Connection String
Provider=OraOLEDB.Oracle.1;Persist Security Info=
Build...
Other Attributes:
OK | Cancel | Apply | Help

توجه إلي اللسان (Record Source) ، واختر الجدول الذي ستعمل عليه ، وهو جدول الموظفين :

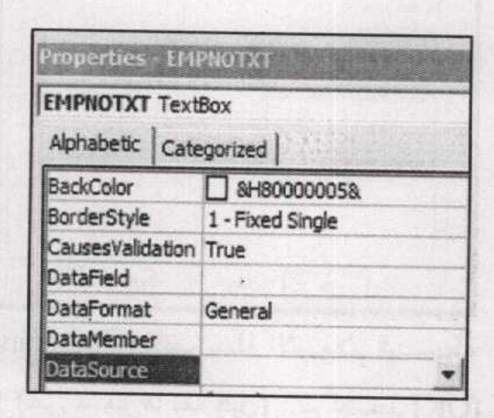


في الخانة (Command) اكتب جملة الاستعلام البـسيطة select * from emp; (التى سنغيرها بعد قليل) : ثم اضغط (OK) :

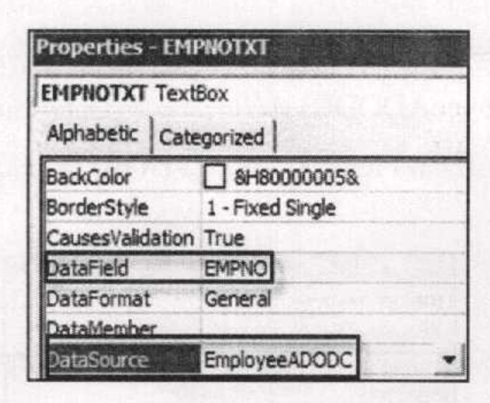


ولنعد الآن الى أدوات النص ، ونركز فى هذا الجانب .

- أمسك بأداة النص (EMPNOTXT) علي النمــوذج ، ثــم توجه إلــي نافــذة الخــصائص . وقــف عنــد الخاصــية DataSource و DataField:



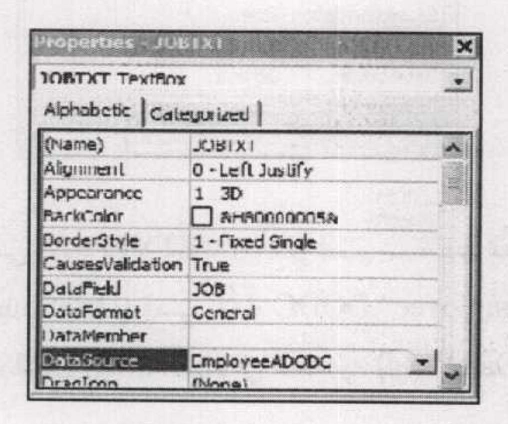
- اضــبط الخاصــية الأولــي (DataSource) إلــي الأداة EmployeeADODC ، ثم اكتب الحقل (EMPNO) فــي الخاصية (DataField) :



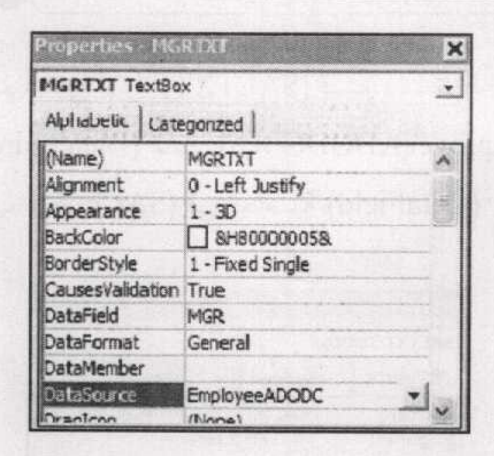
- وفـــي الأداة (ENAMETXT) قـــم بـــضبط الخاصـــية (DataSource) إلـــي الأداة EmployeeADODC ، ثـــم اكتب الحقل (ENAME) في الخاصية (DataField) :



- ثم في الأداة (JOBTXT) قــم بــضبط الخاصــية الأولــي (DataSource) إلـــــي الأداة (EmployeeADODC) ، واكتب الحقل (JOB) في الخاصية (DataField) :



- ثــم فـــي الأداة (MGRTXT) قــم بــضبط الخاصــية (DataSource) إلـــــي الأداة (EmployeeADODC) ، واكتب الحقل (MGR) في الخاصية (DataField) :



- ثم في الأداة HIREDATETXT قـم بـضبط الخاصـية (DataSource) إلي الأداة (EmployeeADODC) ، ثـم اكتب الحقل (HIREDATE) في الخاصية (DataField) :



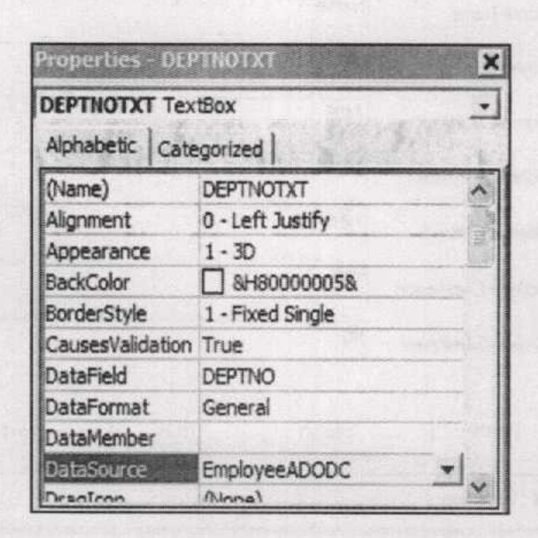
- ثـــم فـــي الأداة (SALTXT) قـــم بـــضبط الخاصـــية (DataSource) إلـــي الأداة EmployeeADODC ، ثـــم اكتب الحقل (SAL) في الخاصية (DataField) :



- ثـــم فـــي الأداة (COMMTXT) قـــم بـــضبط الخاصـــية (DataSource) إلـــي الأداة EmployeeADODC ، ثـــم اكتب الحقل (COMM) في الخاصية (DataField) :



• ثم فـــي الأداة (DEPTNOTXT) قـــم بـــضبط الخاصـــية (DataSource) إلـــي الأداة EmployeeADODC ، ثـــم اكتب الحقل (DEPTNO) في الخاصية (DataField) :



تشغيل التطبيق RUN

بعد أن انتهينا من هذا الجزء فى البرنامج ينبغى أن نـــشاهده وهو يعمل ، للتأكد من نجاح ما قمنا به . وهو ما يمكن تحقيقه مـــن خلال الضغط علي الزر (Run) من شـــريط الأدوات ، أو الـــضغط علي الزر (F5) من لوحة المفاتيح .

وسيظهر لك التطبيق لتجد انه قد تم تحميل البيانات الخاصة بجـــدول الموظفين :

Employee

Employee Number :	7935
Employee Name :	Gamal
Employee Job :	CLERK
Employee's Manager :	7499
Employee's Hiredate :	10/10/2005
Employee's Salary :	1201
Employee's Commission :	500
Employee'sDepartment :	40

Delete | Search | NEW | Save

Employee

قم بتجربة أزرار الأداة (EmployeeADODC) ، التى تمكنك مـن التوجه إلي السجل السابق أو التالي ، وكـذلك إلـى أول الـسجلات وآخرها :

◀◀	◀	Employee	▶	▶▶

وبهذا نكون قد انهينا هذا الجزء ، الذى أثبت لنـا سـهولة الاتـصال بقواعد بيانات أوراكل ، من خلال لغة الفيجوال بيسك 6 .

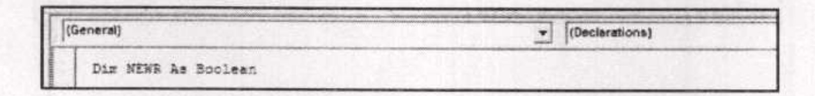
والآن .. سوف نقوم بتصميم ما تبقى من مهام فى هـذا المـشروع ، فنقوم بوضع إجراءات الضغط على كـل مـن : زر الحفـظ ، وزر الإضافة ، وزر البحث ، وزر الحذف..

زر إضافة سجل جديد :

توجه إلي صفحة الكود ، من خـلال الـضغط علـي أي أداة فـى النموذج، وفى أعلي نافذة الكود اكتب الكود التالي :

```
Dim NEWR As Boolean
```

وهذا معناه أننا قمنا بإنشاء متغير باســم (NEWR) ، مــن النــوع (Boolean) ، أي انه سيقوم بتخزين قيمــة مــن (True) أو مــن (False) . وفيما يلى شكل يبين مكان الكود في نافذة الكود لمــن لا يعرف :



الآن.. توجه إلي نافذة تصميم النموذج من جديد (أو اختر الأداة مــن القائمة الموجودة أعلى نافذة الكود) . ثم اضغط (مــرتين) علــى أداة الزر (NEWCMD) ، ثم اكتب الكــود التــالي ، داخــل الإجــراء الخاص بهذه الأداة :

```
NEWR = True
EMPNOTXT.Text = ""
ENAMETXT.Text = ""
JOBTXT.Text = ""
MGRTXT.Text = ""
HIREDATETXT.Text = ""
SALTXT.Text = ""
COMMTXT.Text = ""
DEPTNOTXT.Text = ""
```

زر حفظ التغييرات :

اضغط (مرتين) علي الزر (SAVECMD) ، لكــي تتوجــه إلــي الإجراء الفرعي الخاص بالضغط علي هذه الأداة . ثم اكتــب الكــود التالي :

```
On Error GoTo er:
If NEWR = True Then
rs.AddNew
End If
rs!EMPNO = EMPNOTXT.Text
rs!ENAME = ENAMETXT.Text
rs!JOB = JOBTXT.Text
rs!MGR = MGRTXT.Text
rs!HIREDATE = HIREDATETXT.Text
rs!SAL = SALTXT.Text
rs!COMM = COMMTXT.Text
rs!DEPTNO = DEPTNOTXT.Text
rs.Update
MsgBox "saved Successfully"
NEWR = False
er:
MsgBox "Mission Failed !!!!"
```

زر البحث عن سجل :

اضغط (مرتين) علي الــزر (FINDCME) ، لكــى تتوجــه إلــي الإجراء الفرعي الخاص بالضغط علي هذه الأداة ، ثم اكتــب الكــود التالي :

```
Dim x As String
x = InputBox("what u lookin for ?")
rs.MoveFirst
rs.Find "EMPNO='" + x + "'"
EMPNOTXT.Text = rs!EMPNO
ENAMETXT.Text = rs!ENAME
JOBTXT.Text = rs!JOB
MGRTXT.Text = rs!MGR
HIREDATETXT.Text = rs!HIREDATE
SALTXT.Text = rs!SAL
COMMTXT.Text = rs!COMM
DEPTNOTXT.Text = rs!DEPTNO
```

زر حذف سجل :

الضغط (مرتين) علي الزر (DELETECMD) ، لكى تتوجه إلــي الإجراء الفرعي ، الخاص بالضغط علي هذه الأداة ، ثم اكتب الكــود التالي :

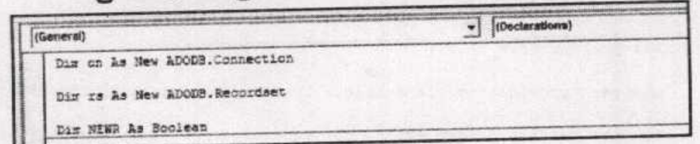
```
Dim x
x = EMPNOTXT.Text
rs.Find "EMPNO='" + x + "'"
rs.Delete
MsgBox "Deleted Successfully"
```

وهكذا نكون قد انتهينا من عملنا ، ولاشك أنه توجد بعض الأخطـــاء والمشاكل التي سنتناولها في صفحات قادمة ، ولكن قبل ذلك علينا أن نشرح – بإختصار – كيفية الاتصال بقاعدة بيانات أوراكل من خلال الكود والمكتبات التي تحدثنا عنها من قبل .

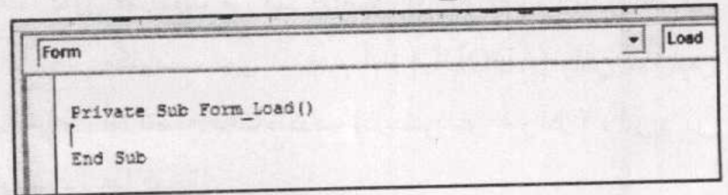
أضف الكود التالى فى جزء من التصريحات العامة :

```
Dim cn As New ADODB.Connection
Dim rs As New ADODB.Recordset
```

وسوف يكون الكود داخل التصريحات كما فى الشكل التالى :



اضغط (مرتين) في أى منطقة فارغة داخل نافذة التـــصميم ، لكـــى نتوجه إلي الإجراء Form_Load :

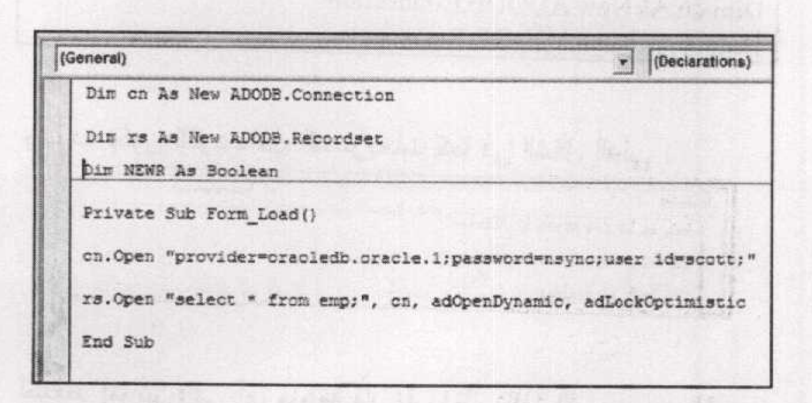


ثم قم بكتابة الكود التالي :

```
cn.Open "provider=oraoledb.oracle.1;password=nsync;user id=scott;"

rs.Open "select *  from emp;", cn, adOpenDynamic, adLockOptimistic
```

ليصبح بهذا الشكل :



وهكذا نكون قد انتهينا من فهم الاتصال بقاعدة البيانات مــن خــلال كتابة كود ومكتبات ، دون استخدام أداة (ADODC) ولكن هذا ليس كل شيء ، لذا ننصح بإقتناء كتاب (كل شئ عن أوراكل) ، لتزيد من خبرتك في هذا المجال.

والآن سوف نناقش إحدى المشاكل ، التى تواجه المستخدم فى كثيــر من الأحيان.

مشكلة NULL :

لكى نفهم المشكلة جيدا سنشرح ذلك بشكل عملى ..

إذن .. قم بتشغيل التطبيق

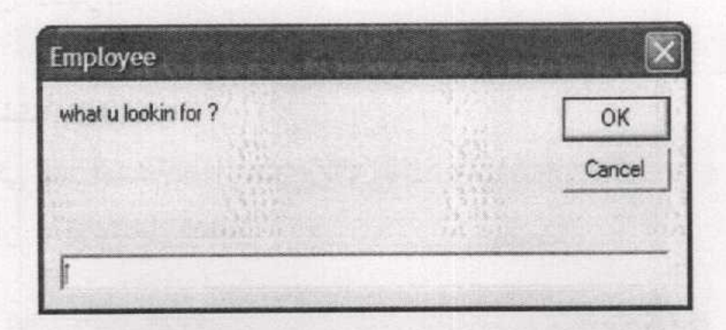
Employee

Employee Number :	7935
Employee Name :	Gamal
Employee Job :	CLEFK
Employee's Manager :	7499
Employee's Hiredate :	10/10/2005
Employee's Salary :	1201
Employee's Commission :	500
Employee'sDepartment :	40

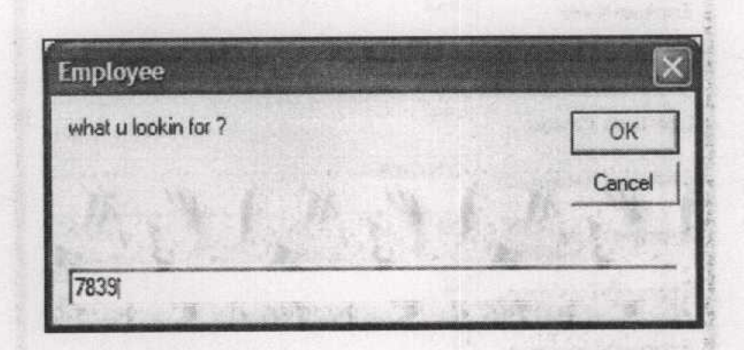
Delete Search NEW Save

Employee

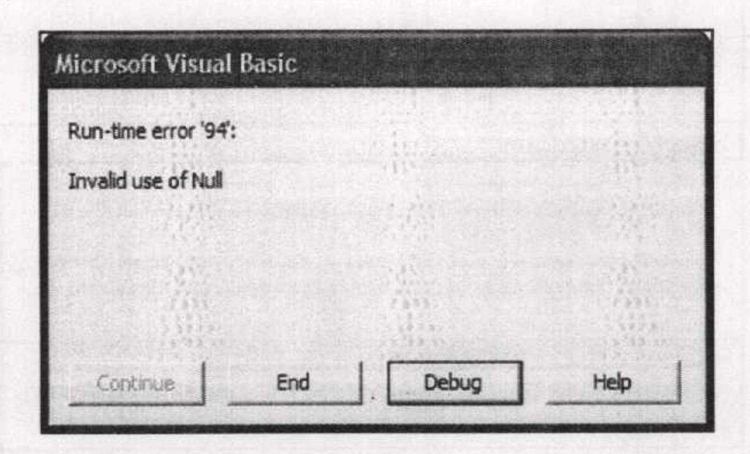
ثم اضغط زر البحث :



ثم اكتب الرقم (7839) ، ثم اضغط (Ok) :



وسوف تفاجئك رسالة الخطأ التالية :



وسبب هذا الخطأ أن الحقل (MGR) يحمل قيمة (NULL) لم نستخدمها استخداماً صحيحاً .

```
JOBTXT.Text = rs!JOB

MGRTXT.Text = rs!MGR

HIREDATETXT.Text = rs!HIREDATE
```

وهذه المشكلة يمكن حلها بعدة طرق ، لكننا نفضل حلها من خلال جملة استعلام سبق استخدامها فى كتابنا هذا ، وهى تتم عبر (NVL) ويمكننا التعويض – من خلالها – عن القيمة (NULL) بأي أرقام أخرى .

أي إننا نريد تحويل جملة الاستعلام التالية :

```
Select * From Emp ;
```

إلي الجملة التالية :

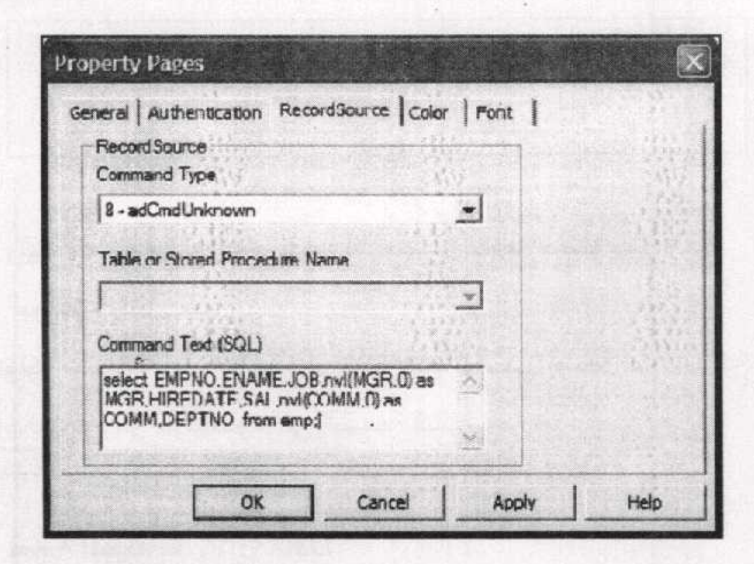
```
select EMPNO,ENAME,JOB,nvl(MGR,0) as
MGR,HIREDATE,SAL,nvl(COMM,0) as COMM,DEPTNO
from emp;
```

وهناك مكانان ينبغى تغيير هذه الجملة فيهما :

الأول : هـو خانـة (Command) فـي نافـذة خـصائص الأداة EmployeeADODC ..

والثاني في الإجراء Form_Load ، عند فـتح مـصدر الـسجلات (RS) .

إذن .. اضـغط بـزر الفـأرة (الأيمـن) علـي الأداة EmployeeADODC ، لتظهر لك نافذة خصائص الأداة ، ثم توجه إلي اللسان Record Source ، واستبدل بالجملـة القديمـة الجملـة الحديثة :



والمكان الثاني الذى ينبغى التغيير فيه هو الإجـــراء Form_Load ، عند فتح مصدر السجلات (RS) :

```
Private Sub Form_Load()
cn.Open "provider=oracledb.oracle.1;password=nsync;user id=scott;"
rs.Open "select * from emp;", cn, adOpenDynamic, adLockOptimistic
End Sub
```

فقم – أيضا – باستبدال الجملة الحديثة بالقديمة:

```
Private Sub Form_Load()
cn.Open "provider=oracledb.oracle.1;password=nsync;user id=scott;"
rs.Open "select EMPNO,ENAME,JOB,nvl(MGR,0) as MGR,HIREDATE,SAL,nvl(COMM,0) as COMM,DEPTNO  f
End Sub
```

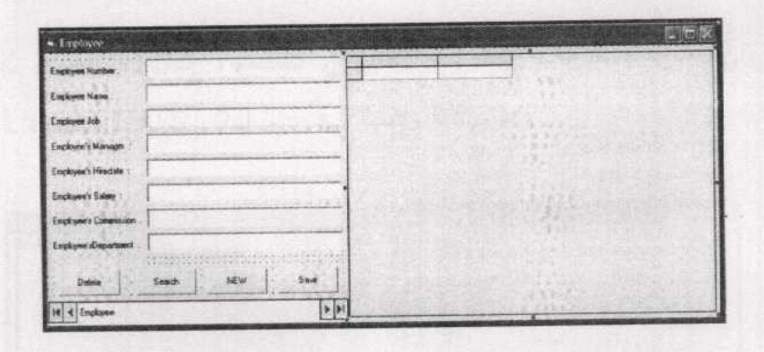
وبهذه الطريقة (العملية) نكون قد صممنا تطبيقا خاصا بإدارة شئون الموظفين بشكل جيد .

ولكن .. بقيت وظيفة واحدة لم نستعرضها ، وهي إضافة أداة شـبكة بيانات . وهو ما سنتناوله في الصفحة التالية .

شبكة البيانات Data Grid

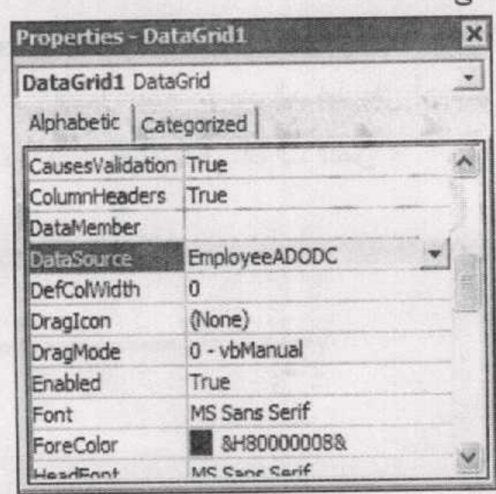
هذه هي الأداة الثانية ، التي ادرجناها من صندوق (Components).

اضغط عليها (مرتين) لإضافتها إلي النموذج ، وحدد موقعها ، كمـا هو واضح من الشكل التالى ، وذلك بعد تكبير حجم النموذج لـيلائم حجم أداة شبكة البيانات:

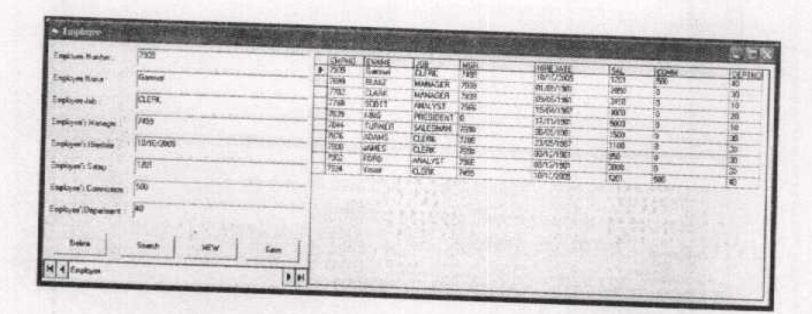


يبقـــــي الآن أن نقـــوم بـــربط الأداة DataGrid بــــالأداة EmployeeADODC

والآن .. توجه إلي خصائص الأداة (DataGrid) ، ثم قـم بـضبط الخاصية (DataSource) إلي EmployeeADODC ، كما تـري في الشكل التالي :



قم بتشغيل البرنامج ، وستجد أن بيانات الموظفين كافة قد ظهرت في أداة شبكة البيانات ، كما ترى فى الشكل التالى :



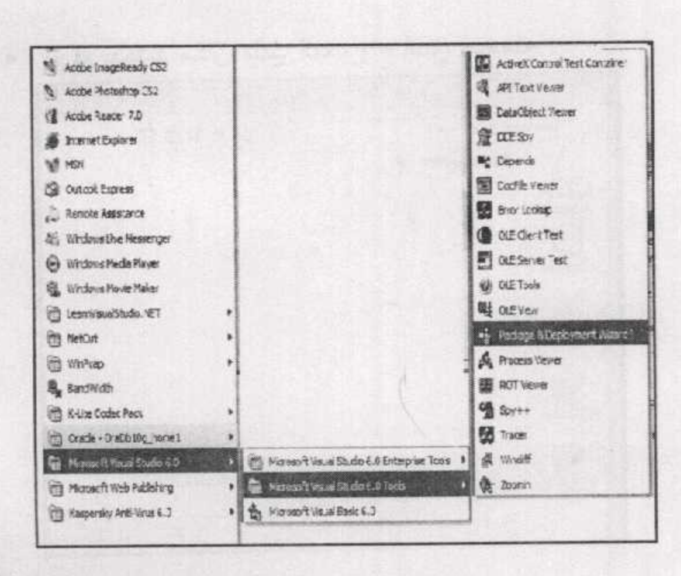
لاحظ .. أن كافة البيانات والسجلات مربوطة ، بعضها بالبعض . لذلك فإنه عند استخدام الأداة EmployeeADODC للتحول من سجل إلي آخر ستجد أن أدوات النص ، وأداة شبكة البيانات تتحرك هى الأخرى إلى السجل نفسه . وعند الضغط على أى سجل في شبكة البيانات ستجدها قد تم عرضها في أدوات النص :

وفي الصفحة التالية سنستعرض كيفية القيـــام بتـــصميم (Package) لمشروعك ، ليعمل علي أي جهاز أخر .

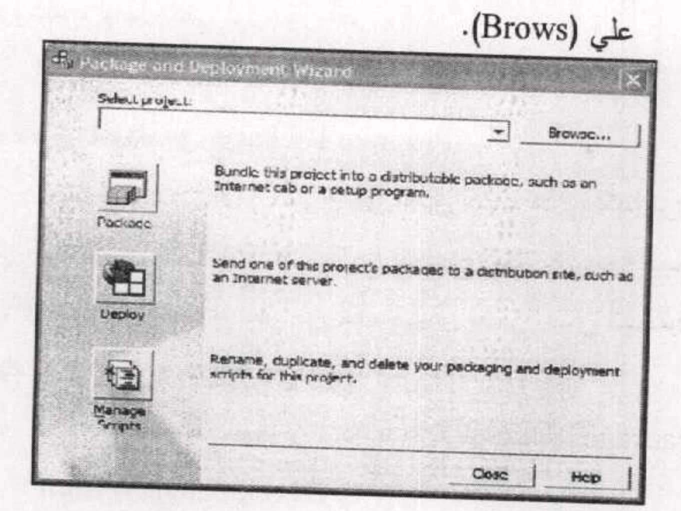
بناء حزمة للتطبيق Program Package

ربما يكون هذا الأمر خارج عن نطاق كتابنا هذا ، لكننا نرى أن من الأفضل معرفة كيفية عمل حزمة لبرنامج قمت بتصميمه مــن خلال الفيجوال بيسك . إذ ليس من المنطقى أن تعجز عــن تــشغيل تطبيقك الذى قمت بتصميمه على جهاز آخر .

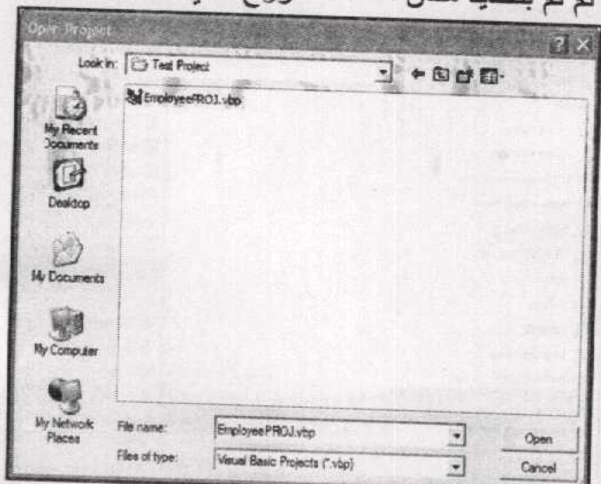
- توجه إلي أدوات الفيجوال أستوديو ، ثم قم بتشغيل Package & Development Wizard :



- وسوف يظهر لك برنامج مصمم حزم التطبيقات ،.ثم الضغط علي (Brows).



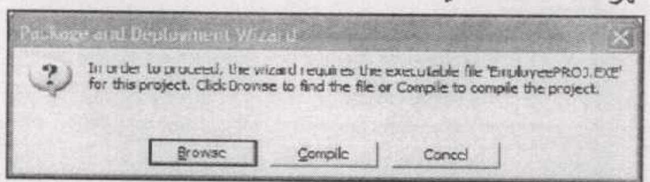
- ثم قم بتحديد مكان ملف المشروع الذي صممته :



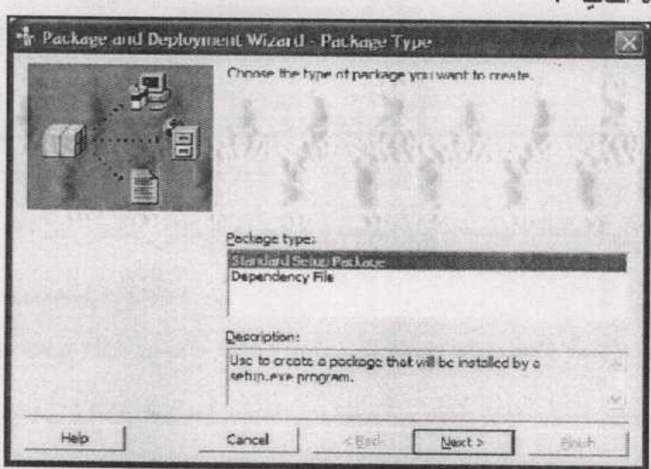
- اضغط الآن علي (Package) :



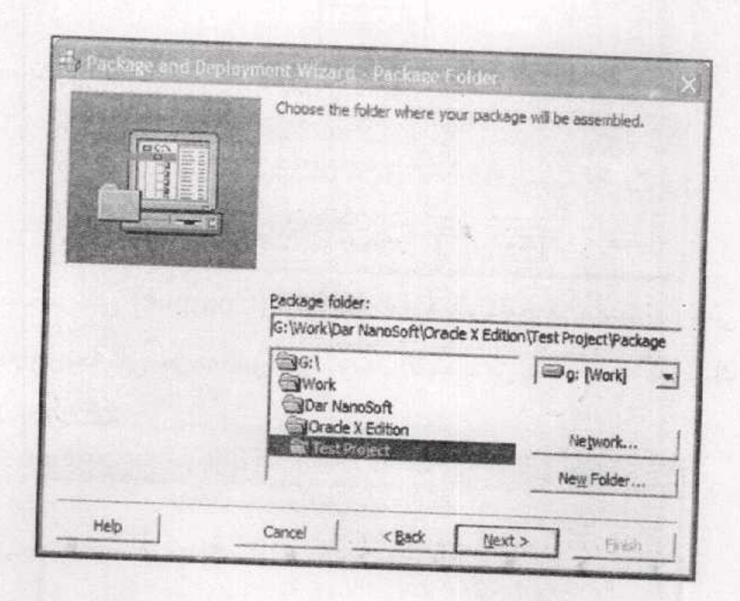
وستظهر لك النافذة التالية :



اضغط علي (Compile) حتى يتم تجميع البرنامج من جديد ، وحتى يتم إدراجه إلي حزمة البرامج . وبعد الانتهاء من ذلك ستظهر لك النافذة التالية :



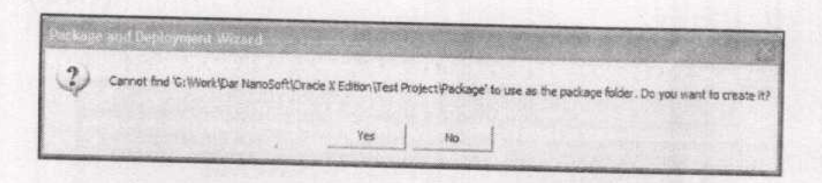
اختر Standard Setup Package من النافذة الـسابقة، واضـغط (Next) ؛ ليتم تحويلك إلى الخطوة التالية ، حيث سيتم تحديد مكـان وضع الحزمة البرمجية:



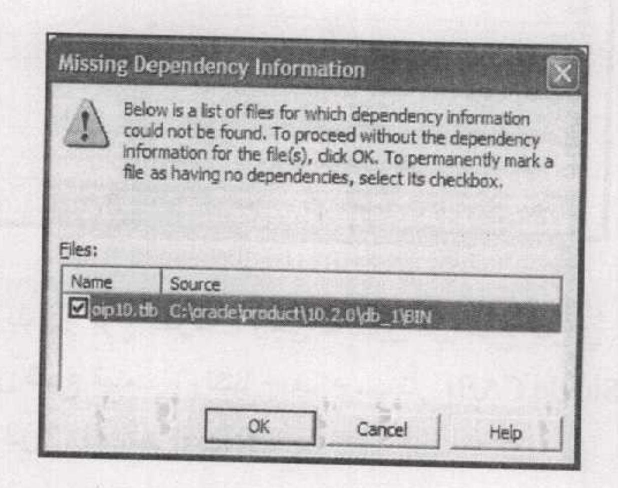
- اختر مكان بناء الحزمة ، وسوف يتم وصفها تلقائياً في مجلد المشروع نفسه .

وربما تظهر لك الرسالة التالية ، التي توضح عدم وجود المجلد الذي ترغب في وضع الحزمة داخله ، وذلك في حالة اختيـارك المكـان الافتراضي .

- إذن .. قم بالضغط علي (Yes) ليتم إنشاء المجلد :

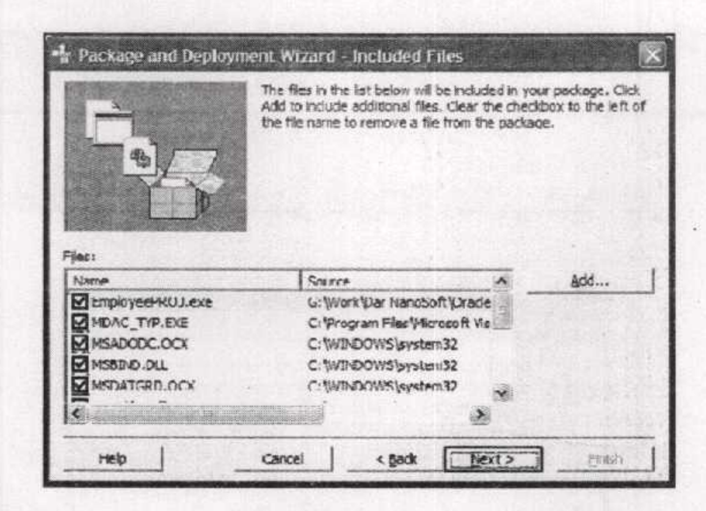


كما أنه أثناء بناء الحزمة ستظهر لك رسالة ، كما فى الشكل التالى :



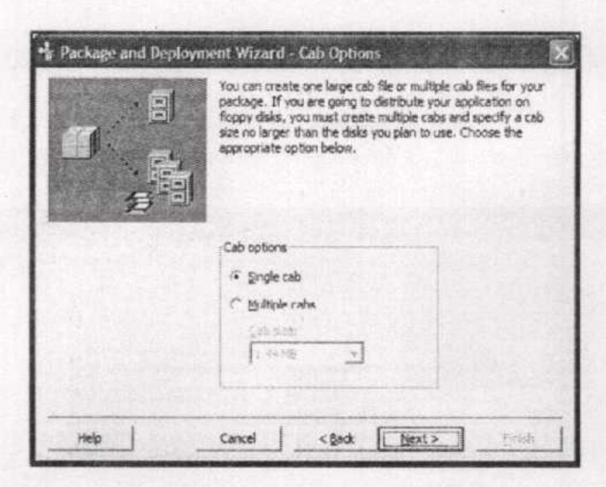
وهى توضح الملفات التي قد يحتاجها التطبيق ، مثل ملفات (OCX) و (DLL) . خاصة بمكتبات هدف البيانات النشط ، أو قاعدة بيانات الأوراكل .

لذا.. اختر أي ملف يظهر أمامك ، ثم اضغط (OK) ، ثـم اضـغط (Next):

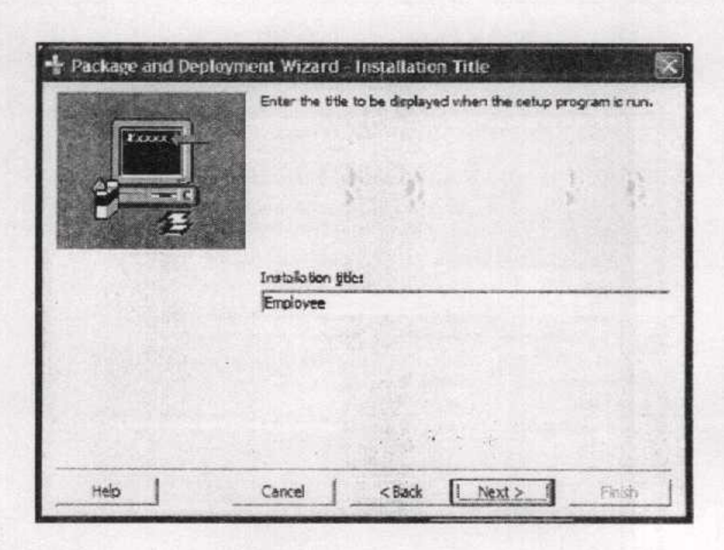


قم الآن باختيار نوع الحزمة ، والتى قد تكون من النوع الفردي ، أو تكون من النوع المتعدد ، ولكننا – هنا – سنختار (Single CAB).

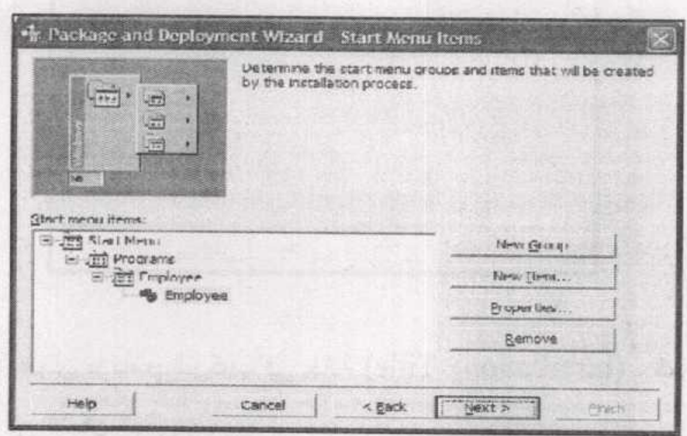
- اذن .. فاضغط (Next) :



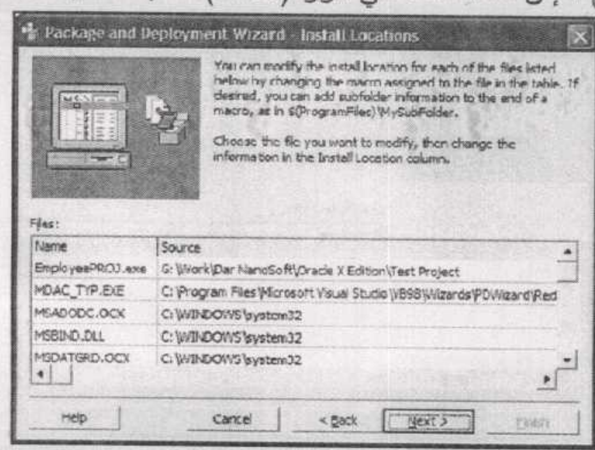
اكتب عنوان برنامج التركيب في خانة (Installation Title) ، كما تري في الشكل التالى :



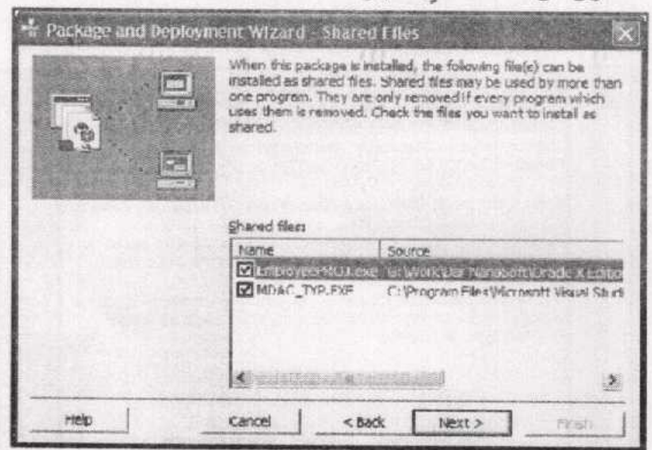
ولتحديد المكان الذي سيتم تركيب البرنامج فيه قم بتحديد هذا ، المكان أو اتركه في المكان الافتراضي ، ثم اضغط (Next) :



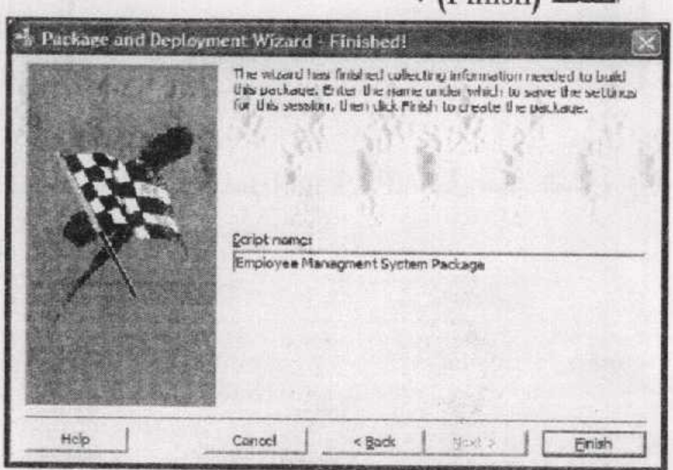
يؤكد البرنامج عى أن مجموعة الملفات تـم إدراجهـا مـن حزمـة البرنامج . إذن .. اضغط علي الزر (Next) لتأكيد العملية :



• الآن .. اضغط علي (Next) في الخطوة (Shared Files)، دون أن تحدث أي تغيير :

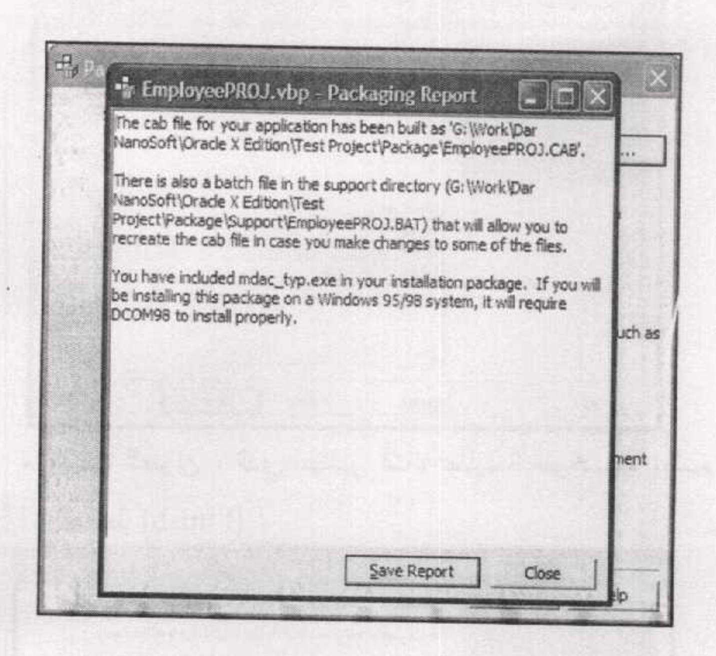


• اكتب العنوان ، الذي سيظهر أثناء عمليـــة التركيـــب ، ثـــم اضغط (Finish) :

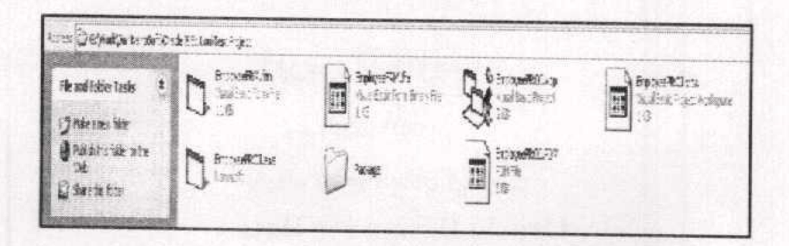


والرسالة التالية تظهر أنك انتهيت بنجاحٍ من هذه العملية .

- إذن .. اضغط (Close) :



عنئذ سيكون لديك مجلد باسم (Package) داخل مجلد المشروع الذي انشأته :



والآن .. يمكنك تركيب البرنامج علي أي نظام آخر ، مادام متــصلا بقاعدة بيانات أوراكل:



وهكذا ننتهى من هذا الكتاب الذى تناولنا فيه الحديث عن مجال قاعدة البيانات أوراكل ، التى يعتبرها البعض مجالا غامــضا ، ونأمــل أن يكون القارىء قد شعر بمدى الجهد الذى بذلناه فى تبــسيط اســلوب الكتاب ، والاستعانة بالأمثلة الكثيرة ، والإنتقال من خطوة إلى أخرى فى سهولة ويسر ، وأن يكون القارىء قد استفاد كامل الاستفادة مــن الكتاب .

وإلى اللقاء فى كتاب قادم إن شاء الله

فهرس المحتويات

الفصل الأول

تصميم قواعد البيانات في أوراكل

Oracle Design for Data

الفصل الثانى

العمل فى بيئة iSQL * PLUS

الفصل الثالث

الأوراكل مع الفيجوال بيسيك 6

Oracle With Visual Basic 6

رقم الإيداع
23536/2007



المركز الرئيسى : 11 شارع د/محمد رأفت – محطة الرمل – الإسكندرية
تليفون وفاكس : 4838326 (03)(2+)
موبايل : 0101634294 (2+) - 0123357844 (2+)
Email: info@egyptbooks.net
URL: www.egyptbooks.net